

# 質問応答と自発的発話の組み合わせによる複数人会話活性化戦略

## Multiparty Conversation Facilitation Strategy Using Combination of Question Answering and Spontaneous Utterance

齋藤彰弘\*<sup>1</sup>  
Akihiro Saito

松山洋一\*<sup>1</sup>  
Yoichi Matsuyama

藤江真也\*<sup>2</sup>  
Shinya Fujie

小林哲則\*<sup>1</sup>  
Tetsunori Kobayashi

\*<sup>1</sup>早稲田大学 情報理工学科

Department of Computer Science and Engineering, Waseda University

\*<sup>2</sup>早稲田大学 高等研究所

Waseda Institute for Advanced Study

Based on our analysis of entertaining multiparty conversations, participants tend to obey the cooperative conversational principle, and provide feedback so as to understand the current topic. In addition, they also speak to express their own idea and interest. In this paper, we describe a conversational robot system designed for constantly facilitating conversations with topic tracing capability, which is able to output a combination of answers to questions and spontaneous utterances. We also describe the database structure of the system and a graphic user interface supporting users to create a large amount of the data.

### 1. はじめに

コミュニケーションロボットを複数人会話に参加させ、会話を活性化するための会話システムを実現するために、質問応答と自発的発話の組み合わせによる会話活性化戦略と発話データベース作成ツールについて検討する。

我々はこれまでに、複数人会話の一種である「難読ゲーム」にロボットを参加させ、その会話を活性化させる取り組みを行ってきた [谷山 10]。通所介護施設におけるゲームに構築したロボットを実際に参加させ、活性化の効果が見られることを確認した [Matsuyama 10]。ロボットは場を活性化させるために、回答者である高齢者に回答を促したり、自ら回答したり、問題に関する発話をしたりする。その中で、ロボットがそれまでの会話の流れ、すなわち文脈を踏まえた上で会話参加者の興味を引く発話をする事で、会話が大きく活性化することが分かった。難読ゲームの中でこのような発話を出力する枠組みを提案する。

本研究では、ロボットが発話する内容は、発話データとして予め人手でデータベースに入力しておく方法を取る。その中で、非常に興味を引く内容を持った発話データが作成されることがあるが、実際にロボットが発話する際は、それらを文脈を踏まえて発話することが必要であることが実験から分かっている。通常の会話システムでは、人の発話内容に応じてロボットが発話を決める質問応答システム（例えばイフボット [伊吹 05] など）のようなものが多い。しかし文脈を踏まえた上で特定の内容を持った発話をするためには、ロボット自ら文脈を制御できる枠組みの導入が必要である。そのため、ロボットに質問応答だけでなく、文脈を制御するための自発的発話を行わせる枠組みを提案する。

文脈を踏まえた発話をするため、文脈を発話内容そのものの履歴ではなく、発話内容を抽象化した「トピック」の系列で表現する。ある発話をロボットがすると、その発話データに結び付けられたトピックが文脈として展開される。ロボットは、展開されたトピックに関連した発話の中から次の発話を選択す

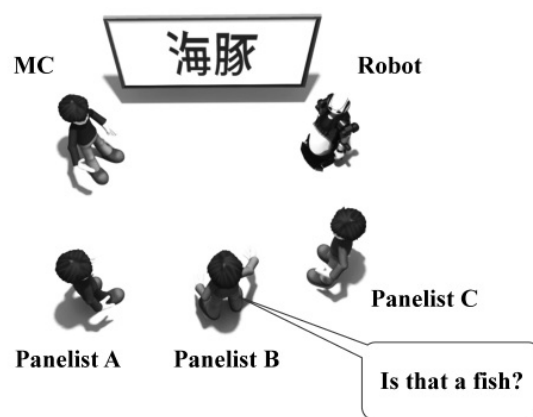


図 1: 難読ゲーム

ることになる。前述の通り発話データは人手で入力するが、発話データ作成者は、トピック同士の繋がりや、どのトピックに興味を引く発話内容があるかなどを常に把握しながら入力をする必要がある。しかし発話データが膨大であるため、把握することだけでさえ困難であり、非常に時間と手間がかかる作業となっている。本研究では、トピック同士の繋がりを視覚的に表示することで発話データの作成を支援するツールを開発する。

### 2. 会話活性化戦略

#### 2.1 会話分析

難読ゲームとは、図 1 のようにプロジェクトで投影された読むことが難しい漢字の読みを答えるゲームである。ゲームには司会者と複数の回答者が存在する。そこにロボットが回答者の一人として参加し、漢字の問題に関する回答やヒント、また問題に関連したトピックの発話を行ったりする。

通所介護施設の実験において、ロボットの特定の発話に参加者の興味を引き、場が活性化される様子が観察された。その会話の例を図 2 に示す。ロボット (Robot) の発話「僕の好きな女優はね、オードリー・ヘップバーンだよ。ローマの休日はいい映画だよね。」の直後に歓声が上がった。直前の司会者 (MC)

連絡先: 早稲田大学基幹理工学部情報理工学科  
知覚情報システム研究室  
〒 169-8555 東京都新宿区大久保 3-4-1  
55 号館 N 棟 5 階 509 号室  
E-mail: saito@pcl.cs.waseda.ac.jp

表 1: 会話の実行例

	発話内容	現在のトピック	期待される次のトピック
Robot	「ハリウッドといえば、映画だね」	—	映画, 俳優
MC	「好きな映画は何？」		
Robot	「ローマの休日だよ」(a) 「オードリーが出演しているから」(b)	映画 ローマの休日	ローマの休日, 映画, 俳優 オードリー・ヘップバーン, ローマの休日, 映画, 俳優
MC	「オードリーをどう思う？」		
Robot	「妖精みたいでしょう」(c)	オードリー・ヘップバーン	オードリー・ヘップバーン, ローマの休日, 映画, 俳優

表 2: 会話の実行例で利用した発話データ

	発話データ (a)	発話データ (b)	発話データ (c)
Topic	映画	ローマの休日	オードリー・ヘップバーン
Predicate	好き	好き	思う
Question Type	What	Why	How
Response	ローマの休日だよ	—	妖精みたいでしょう
Active Response	—	オードリーが出演しているから	—
Next Topic	ローマの休日	オードリー・ヘップバーン	—

	(問題「ハリウッド」出題中)
MC	「正解です。ロビスケは何か好きな映画はあるの？」
Robot	「僕はねタイタニックが好きだったなあ。」
MC	「誰か好きな俳優さんはいるの？ロビスケ」
Robot	「(A) 僕の好きな女優はねオードリー・ヘップバーンだよ。(B) ローマの休日はいい映画だね。」 (一同歓声)

図 2: 活性化した会話例

の質問は、ロボットの好きな俳優が誰か問うものであり、俳優の名前を回答したことで要求は満足された (図 2 の (A))。ロボットはその情報に加えてその俳優に関連する映画について発話した (図 2 の (B))。この発話は参加者が意図しなかったものである。文脈を踏まえた上での意外性のあるこの発話が場を活性化させたと考える。

以上より、単に質問に回答する発話をするだけでなく、参加者の興味を引く発話をするのが活性化の一因になると言える。また、興味を引く内容は、その場での会話の内容、すなわち文脈を踏まえて発話することも重要であると考えられる。

## 2.2 トピックによる文脈表現とその制御

文脈を踏まえて発話するためには、文脈を正確に把握する必要がある。文脈という言葉は広い概念であるが、本研究では発話内容の履歴であるとする。例えば、図 3 において「オードリー・ヘップバーンが出演しているからね」という発話は、その前の「好きな映画は何?」「ローマの休日だよ」という発話の履歴を踏まえた発話であると考えられる。

発話の表層表現そのものの履歴を文脈として用いることも考えられるが、図 2 と図 3 の例を見て分かるように、同じオードリー・ヘップバーンを話題とした発話が、異なる発話の履歴によって生成されることもある。これを実現すれば、興味を引く発話を行う機会が増える。そこで本研究では、文脈を発話内容そのものではなく、発話内容を抽象化した「トピック」というものを導入し、その履歴で表現する。図 3 の例では、「ローマ

トピック遷移:「映画」→「ローマの休日」→「オードリー・ヘップバーン」

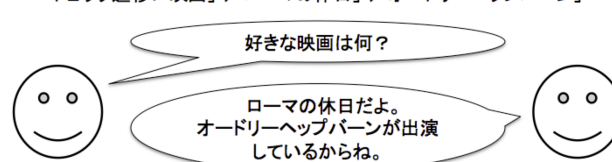


図 3: トピックの遷移

の休日だよ」という発話は「映画」というトピックに関するもので、文脈に「ローマの休日」という別のトピックを付け加える。「オードリー・ヘップバーンが出演しているからね」という発話は同様に、「ローマの休日」というトピックに関するもので、文脈に「オードリー・ヘップバーン」というトピックを付け加える。

トピックの履歴を文脈として用いることで、文脈を踏まえる発話を行うことができるが、その上で興味を引く発話をするためには、その発話に繋がるようにシステム自ら文脈を制御しなければならない。そうでなければ、質問者が都合よく興味を引く発話を引き出す質問をすることを待たなければならないからである。そのため、本研究では質問されたときに回答するだけでなく、自発的に発話を行い、興味を引く発話につながるようにトピックを繋げていくことができる枠組みを検討する。

## 2.3 会話システム設計

文脈を踏まえた上で興味を引く発話をするために、質問応答と自発的発話を組み合わせる方法について検討する。質問応答、自発的発話に関わらず、発話の一つのトピックと関連する。例えば、「僕の好きな女優はねオードリー・ヘップバーンだよ」という発話は、「俳優」というトピックに関連する発話である。質問応答の場合、この発話は「好きな俳優は誰ですか?」という質問に対する回答になるが、「俳優」というトピックが文脈として保持されていれば「誰が好きな?」のように明示的に「俳優」という言葉が質問に含まれていなくてもその回答を選択することができる。一方、自発的発話の場合は「俳優」と

いう文脈がある場合に限り、この発話を選択することになる。

以上のような仕組みを実現するために、個々のシステム発話は以下の項目を持つ発話データとして表現される。

1. Topic : トピック名
2. Predicate : 述部 (動詞と形容詞)
3. Question Type : 質問タイプ (6W1H)
4. Response : 質問応答の発話内容
5. Active Response : 自発的発話の発話内容
6. Next Topic : 期待される次のトピックに加えるトピック名

Topic は、先の例の「俳優」に相当するもので、発話が生成される際に文脈が持つべきトピックを表している。Predicate, Question Type は、質問応答をする際に質問者の発話が満たすべき条件を示している。Predicate は、好き、見る、などの述語に相当するもので、Question Type は、誰、どこ、などの6W1Hに相当するものである。例えば、「好きな俳優は誰ですか?」という質問の場合、Predicate は「好き」で、Question Type は「who」となる。Response, Active Response は、それぞれ質問応答の場合の応答、自発的発話の際の発話内容を表す。Next Topic は、発話を行った場合に展開される次のトピックを表す。例えば、「僕の好きな俳優はねオードリー・ヘップバーンだよ。」という発話に関しては、「オードリー・ヘップバーン」というトピックが設定される。

個々の発話が以上のような発話データで表現されることで、以下の流れで会話が進行する。まず、文脈が保持するトピックを Topic に持つ発話が次の発話として限定される。ここで、質問者がある質問をした場合は、その発話の音声認識結果が、Predicate に相当する単語や、Question Type に相当する代名詞 (who であれば「誰」など) を持つかどうかをチェックし、さらに発話を限定する。その結果、いくつかの発話が残った場合は、その中からランダムに発話を選択して、Response の内容を発話する。質問者による質問が一定時間無ければ、限定された発話の中から一つをランダムに選択して、Active Response の内容を発話する。発話が行われた後は、Next Topic に書かれたトピックを文脈に加える。このように、システムは自らの発話によって次々とトピックを展開する、すなわち次のトピックとなる候補を増やすことで会話を進行し、興味を引く発話に会話の参加者を引きこんでいく。

表1において例を示す。「好きな映画は何?」と質問されると、「好き」という Predicate, 「何」という Question Type を基にして、表2に示されている発話データ (a) が出力され、期待される次のトピックに「ローマの休日」が加わる。ロボットは「ローマの休日だよ」と質問応答した直後に自発的発話を行うために、期待される次のトピックの中で後に加わったもの順に探索し、表2における「ローマの休日」の発話データ (b) を出力した。

### 3. 発話データ作成支援ツール

#### 3.1 発話データ作成とその問題点

会話活性化を実現するための、興味を引く発話をするためには、2.3で述べた発話データを豊富に用意しておく必要がある。そのため、本研究では発話データを事前に人手でデータベース内に入力しておく方法を取る。

各発話データは、トピックを手がかりにして関連性を持つものの、実際の会話においてはいわゆるスクリプトのように決められた順番に生成されるものではない。従って、個々の発話データは、別々に作成されることとなり、別々の時間に、また場合によっては複数の別の作業員によって作成される可能性もある。このような状況で一貫性を保った発話内容を持つデータを作成するためには、それまで入力された発話内容を把握しておくことが必要となる。しかしながら、全ての発話データを閲覧してその全ての内容を把握することは、発話データの量が多くなると事実上不可能となる。

また、発話データを作成する際、発話内容が先行する場合とトピックが先行する場合の二つの場合が考えられる。例えば、文脈に「映画」というトピックが既にあることを想定して、関連する発話内容である「好きな映画はローマの休日だよ」を考え、その後で「映画」→「ローマの休日」というトピック遷移を考えることがある。これは、発話内容を先に考えることから発話内容が先行する例と言える。一方、例えば、「オードリー・ヘップバーンはいい女優だね」といった発話内容が、特に文脈とは関係なく作成されている場合を考えると、その発話に到達するためのトピック遷移を先に考え、その遷移を実際に実現するために後から関連する発話内容を考えることになる。これは、トピックを先に考えることからトピックが先行する例と言える。本研究ではそれぞれの発話データ作成スタイルについて、前者を「発話内容先行型」、後者を「トピック先行型」と呼ぶ。

発話内容先行型は、既にあるトピックに対して何らかの発話内容を考えることになる。発話内容としては Wikipedia 等の辞書的网站から得られる豆知識や、ちょっとした面白いフレーズなどを考える必要があるが、その作業はある一つのトピックに注目していれば十分であるためそれほど難しいものではない。トピック先行型は、目的の発話を生成するためのトピック遷移を実現するために、複数のトピックを具体的な発話内容無しに作成することになる。この作成スタイルを取ることで、目的の発話に至るための複数のトピック遷移を作成した上で、特に効率的に遷移する系列を選択して、それぞれのトピックに対して具体的な発話内容を作成するといった順で作業が進められる。これにより、無駄な発話内容の作成を避けられ、全体としては効率的に作業が進められると考えられる。しかしながら、複数の系列の効率性を比較するためには、途中で幾つのトピックが存在するかや、それぞれのトピックの発話内容が考えやすそうかなどの情報が必要である。これらの情報を、個々の発話データを表の形で見る従来のデータベースのインターフェースから得ることは非常に困難である。

#### 3.2 発話データ作成支援方法

前節で述べたように、発話データ作成における多くの問題は、発話データが大量になったときに、そのデータ同士の関連性、特にそのトピック同士の関連性が把握しにくいことに起因する。そのため、トピックをノードとし、関連のあるノード同士をアークで結んだネットワークとして表示することで閲覧をやすくする。しかし、このような方法を取っても、データが大量になれば表示は複雑になり、把握は難しくなる。そこで、あるノードを選択した上で、そのノードに対して関連するノードのみを整理させる機能を持たせる。

ノードを選択すると、該当トピックに既に作成された発話データ一覧を表示する。新しい発話内容の作成は、この状態で入力を行う。Wikipedia における該当トピックの項目を表示する機能を持たせることで、作業員の発想を支援する。

新しいトピックの作成は、ネットワークにノードを加える



図 4: 発話データ作成支援ツール

という作業によって簡単にできる。本研究のシステムでは、トピックが遷移するには必ず発話を伴うため、発話内容が無いトピック遷移は「仮リンク」として表示することにする。通常のアークと、仮リンクを区別して表示することで、トピック先行型の発話データ作成スタイルにおいて、どのリンクに対して発話内容を作成すべきかが容易に分かるようになる。

### 3.3 ツールの実装

発話データ作成支援ツールを、データベースと連携する Web インターフェースとして実装した。図 4 にその様子を示す。3.2 で述べたように、各ノードが一つのトピックに対応し、関連があるトピック同士はそのノードがアークによって結ばれている。また、トピック同士の関連性は、次のトピックが展開するという方向性を持ったものなので、有向アークで表示している。各ノードはドラックによる移動が可能で、作業者の見やすいように自由に配置を変えることが可能である。また、ノードを選択した上でツール上のボタンを押すことで、そのノードに関連したノードのみを表示することもできる。

また、発話内容の作成に関しては、ノードを選択した上で表示されるフォーム内で作業を行う。図 4 は、トピック「映画」に対応するノードを選択した状態を示している。発話内容の追加・修正・削除をこのフォーム内で行うことができる。また、トピックに関する Wikipedia の項目の表示機能や、Web 検索機能があり、発話内容を作成する際に適宜参照できる。

データベース内に存在するトピックの数が多くなると、既存のトピックにどのようなものがあるか把握しづらくなり、同じ、あるいは同じような意味を持ったトピックを複数作成してしまう可能性がある。この問題を解消するため、あるトピックが既に存在するかどうかを確認するための検索機能を実現した。また、発話内容作成において Next Topic を指定する際に、入力部分文字列と一致する名前を持つトピックが既に存在している場合は、リストにして表示することにより、重複した作成を防いでいる。

トピック先行型の発話データ作成スタイルを実現するために、特に他のトピックと関連を持たないノードを作成する機能、またノード同士を具体的な発話内容無しに関連付ける機能を実現した。この際、具体的な発話内容が入力されるまでは実際の会話ではこの遷移を辿ることが出来ない。そこで、この繋がりを「仮リンク」と呼び、通常のアークとは異なる形で表

示することにした。発話内容がある通常のアークは黒い色で、仮リンクは赤い色で表示される。仮リンクを見つけやすくすることによって、効果的な発話に繋がる仮リンクに対して優先的に発話内容を付与することができる。こうすることで、効果的な発話が生成されやすくなるように、効率的にデータを修正することが可能なる。

## 4. まとめ

本研究では、会話を活性化させる発話とは発話内容自体が興味を引くだけでなく、その発話が前提として会話の文脈を踏まえたものであるべきであると考えた。そこで会話の文脈を踏まえた質問応答をするだけでなく、ロボット自身が興味を引く発話を行うために文脈を制御する自発的発話を行うシステムを提案した。人間同士の会話にこのロボットが参加することで、活発な会話の実現できると期待される。また、既に入力された膨大な発話データにおける会話の流れの把握などが原因で発話データの作成が困難だった問題を解決する支援ツールを開発した。このツールによって発話データの作成する量が増えることに加え、会話の文脈を踏まえた発話内容を作成しやすくなり、会話活性化に貢献できると期待される。今後は提案した会話システム、発話データ作成支援ツールを利用して対話実験を行い、更に会話活性化の要件を検討していく予定である。

## 参考文献

- [伊吹 05] 伊吹征太, 木村憲次, 武田夏佳: コミュニケーションロボットを用いた高齢者生活支援システム, 日本機械学会誌, 日本機械学会, pp.392-395(2005).
- [谷山 10] 谷山輝, 松山洋一, 藤江真也, 小林哲則: 参与構造を考慮した行動設計に基づくグループ会話ロボットの構築, 人工知能学会言語・音声理解と対話処理研究会, pp.55-60(2010).
- [Matsuyama 10] Yoichi Matsuyama, Shinya Fujie, and Tetsunori Kobayashi: Framework of Communication Activation Robot Participating in Multiparty Conversation, AAAI Fall Symposium, Dialog with Robots, pp.68-73(2010).