

# オンライン上における大規模議論支援のための IBIS構造に基づく自動ファシリテータの実装

柴田 大地<sup>1,a)</sup> 西田 智裕<sup>1,b)</sup> 山口 直子<sup>1,c)</sup> 鈴木 祥太<sup>1,d)</sup> 芳野 魁<sup>1,e)</sup> 平石 健太郎<sup>1,f)</sup>  
伊藤 孝行<sup>1,g)</sup>

**概要:** 多様な意見を開かれた形で集めることができるため、オンラインにおける議論が注目されている。議論においてファシリテータは重要な役割を果たしているが、大規模な議論に向けて、人間のファシリテータにはバイアスや時間的な面で限界がある。従ってファシリテータを自動化することが求められている。そこで私たちは議論支援を行う自動ファシリテータを提案する。提案するファシリテータはIBISに基づき議論を構造化した上で、適切に議論が進むような発言を生成する。評価実験では本研究で提案するファシリテータを実装し、実際に議論の支援を行なった。実験結果から議論の進行支援が可能であることを確認した。

## Supporting Large-Scale Online Discussion by Automated Facilitator based on IBIS

**Abstract:** Discussion support systems on the Web have attracted attention because various opinions can be discussed openly. In this regard, the facilitator plays an important role for leading various participants in order to make constructive discussions on the Web. However, human facilitators for large-scale Web discussions have limitations in terms of their resources. Therefore, this paper proposes a novel automated facilitator approach for supporting large-scale online discussions. Specifically, the proposed facilitator agent models online discussions with issue based information system (IBIS) structure in order to make these discussions easy to understand for both humans and intelligent agents. Experimental results demonstrated the efficiency of the proposed automated facilitator in promoting the progress of large-scale online discussions.

### 1. はじめに

多様な意見を開かれた形で議論が可能のため、オンラインにおける議論支援に注目が集まっている [1] [2] [3]。オンラインにおける議論では参加者が多様な意見を述べるのが重要である。しかし、従来の掲示板のような仕組みは議論を行うことに適していない。理由は大きく2つ存在し、1つは議論の方向性がまとまらず、議論が錯綜してしまうという点、2つ目は他者を侮蔑するような発言、いわゆる炎

上 [4] が起こりうる点である。以上の理由から、オンラインにおいて参加者が建設的な議論を行うことは難しい。

オンラインにおける議論の質を上げるために有効なアプローチの1つとして、ファシリテータの導入が挙げられる [5] [6]。ファシリテータとは会議やプロジェクトなどの集団活動がスムーズに進み、成果が上がるように支援することを専門に担当する人のことである。ファシリテータ自身は集団活動そのものに参加せず、参加者の意見を引き出し、議論や意見集約を適切にリードする役割を持つ。一般市民による議論、コンセンサス会議、および科学技術分野のワークショップなど、広い分野でファシリテータの役割が注目されている。

ファシリテータを中心とした議論支援システムとして、伊藤らの研究 [7] が挙げられる。伊藤らの研究では、ファシリテータがワークショップや社会実験の議論支援を行なった。実験では参加者が他者を中傷するような現象は見られ

<sup>1</sup> 名古屋工業大学大学院情報工学専攻  
Nagoya Institute of Technology, Gokiso, Showaku, Nagoya  
466-8555, Japan  
a) shibata.daichi@itolab.nitech.ac.jp  
b) nishida.tomohiro@nitech.ac.jp  
c) yamaguchi.naoko@nitech.ac.jp  
d) suzuki.shota@itolab.nitech.ac.jp  
e) yoshino.kai@itolab.nitech.ac.jp  
f) hiraiishi.kentaro@itolab.nitech.ac.jp  
g) ito.takayuki@nitech.ac.jp

れず、ファシリテータの社会的な存在感 [8] [9] が議論に影響を与えた可能性がある。一方で人間のファシリテータには大規模な議論支援を行う上で、バイアス、時間的、そして能力的な面で限界がある。議論が大規模または長期に渡る場合、議論をコントロールし、合意に導くためにはファシリテータの自動化が必要不可欠となる。

ファシリテータの自動化に向けて様々な取り組みがなされている [10] [11] [12] [13]。しかしファシリテータが参加者の発言を理解し、発言を行うような取り組みは少ない。そこで本研究では議論を構造化することで、自動ファシリテーションの質を向上させるような自動ファシリテータエージェントを提案する。

議論を構造化するための手法として、IBIS 構造 [14] を用いる。議論マイニング [15] [16] とは課題を明示的に含む点で大きく異なる。課題をもとに議論を進めることで分かりやすい議論が可能となる。

提案手法である自動ファシリテータの動作を図 1 に示す。参加者が議論支援システムに投稿 A, B を行う。A: ”河川の氾濫に対応するにはどうすれば良いか?”, B: ”センサーネットワークなどの導入が必要だと思います”。マネージャーエージェントは議論支援システムから議論のログを取得し、議論抽出を行うようにリクエストを送る。リクエストを受け取った議論構造抽出エージェントはノード抽出とリンク抽出を行う。ノード抽出では A が Issue, B が Idea であると抽出を行い、リンク抽出では A と B 間に B responds to A という関係があることを抽出する。次に、発言生成エージェントが発言候補を生成する。発言生成の方法の一例として、図 1 (b) にあるように、Issue が来たら Idea をリクエストするような発言候補を作る。最後にマネージャーエージェントが発言候補から投稿を選び、議論支援システムに投稿を行う。

本稿では第 2 章で本研究を説明するために必要となる関連研究について述べる。第 3 章で議論の構造抽出について説明し、第 4 章では IBIS 構造に基づく自動ファシリテータについて説明する。第 5 章で自動ファシリテータの有効性評価のための実験の概要と結果について述べる。最後に第 6 章で本論文についてのまとめを述べる。

## 2. 関連研究

### 2.1 IBIS 構造

IBIS 構造 (Issue-Based information system) [14] とは議論を構造化するための手法で、議論構造を理解しやすい形式にすることで複雑な問題に対し、意思決定を助けることを可能とする。IBIS 構造の要素は大きく 3 つに分類される。

**Issue:** 解決すべき課題。

**Idea:** 課題に対する解決案。

**Argument:** 解決案に対する意見。

Argument はさらに Pros (賛成意見), Cons (反対意見)

の 2 つに分類できる。IBIS 構造の要素をノード、要素同士の関係をリンクと呼ぶ。

### 2.2 D-Agree

D-Agree とは、大規模合意形成システム COLLAGREE [17] に API 機能を拡張したシステムである。電子掲示板型の自由な投稿システムを基本としたシステムであり、誰でも議論に参加することが可能である。D-Agree のシステムは PHP で記述されており、Web API 機能が実装されているため、外部プログラムと容易に連携することが可能である。

## 3. 議論の構造抽出

IBIS 構造 [14] に則り議論の構造推定を行う。議論支援システム上の投稿に対し、構造推定をするために、本研究では議論を構造化するための課題をノード抽出とリンク抽出の 2 つに分ける。

**ノード抽出:** 投稿を 1 文に分け、IBIS 構造の要素 (Issue, Idea, Pros, Cons) に分類する。1 文と IBIS 構造の要素は 1 対であるとする。

**リンク抽出:** ノード抽出を行った要素間のリンク関係を抽出する。IBIS 構造を仮定しており、主要なリンクである Issue-Idea, Idea-Pros, および Idea-Cons 間のリンク抽出を行う。

### 3.1 ノード抽出モデル

本研究ではノードの抽出モデルに Doc2Vec [18] と cos 類似度を用いた。生成したモデルを用いて、IBIS 構造の要素との類似度を計算する。Issue に関する投稿から生成したモデルを用いると Issue の代表ベクトルを得ることができる。同様に議論データから Idea, Pros, Cons の代表ベクトルを得る。そして Doc2Vec のモデルから投稿中の文章をベクトル化する。

代表ベクトルとベクトル化された投稿の cos 類似度を求めることで、ベクトル間の類似度を求めることが可能となる。

cos 類似度は式 2 で求めることが可能である。

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} \quad (1)$$

$$= \frac{\sum_{i=1}^{|\mathcal{V}|} q_i d_i}{\sqrt{\sum_{i=1}^{|\mathcal{V}|} q_i^2} \sqrt{\sum_{i=1}^{|\mathcal{V}|} d_i^2}} \quad (2)$$

$q, d$  はそれぞれ  $V$  次元のベクトルである。

似ている投稿であれば同じクラスに分類されると仮定し、文章を類似度の最も高いクラスに分類する。最後に各クラスの類似度を softmax 関数にかける。softmax 関数は式 3 で表される。

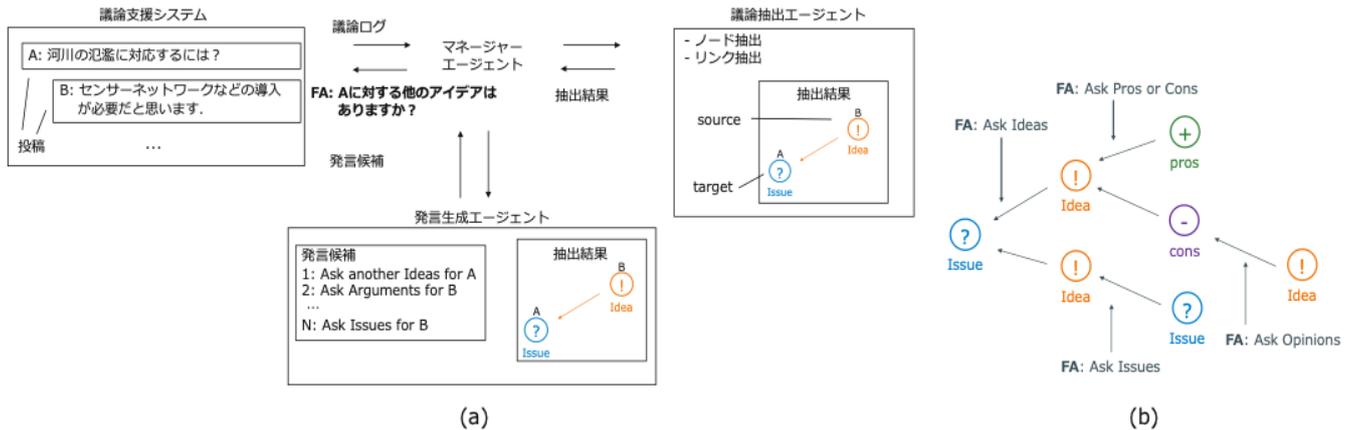


図 1 提案手法の概要. (a) 自動ファシリテータエージェントの動作例. (b) IBIS 構造に基づくファシリテータの発言生成手法.

$$y_i = \frac{\exp(a_i)}{\sum_j^D \exp(a_j)} \quad (3)$$

$y_i$ ,  $a_i$ , および  $D$  はそれぞれ  $i$  番目の出力,  $i$  番目の入力, および要素の数を示す. softmax 関数に通すことで, 各クラスの類似度を合計すると 1 となるため確率として扱うことが可能となる.

### 3.2 リンク抽出モデル

リンク抽出モデルには本研究プロジェクトで開発されているモデルを用いる. fastText [19] で単語を分散表現に直し, Bidirectional LSTM モデル [20] [21] を用いてリンクの抽出を行うモデルである.

## 4. IBIS 構造に基づく自動ファシリテータ

本章で, 自動ファシリテータエージェントを実現するための提案手法を説明する. 提案する自動ファシリテータはマネージャーエージェント, 発言生成エージェント, 議論構造抽出エージェントの 3 つのエージェントから構成される.

### 4.1 マネージャーエージェント

マネージャーエージェントは議論構造抽出エージェントと発言生成エージェントを管理し, 実際に議論支援システムに投稿を行うエージェントである. 議論構造抽出エージェントから議論構造の推定結果を受け取り, 受け取った結果を元に発言生成エージェントへ発言の生成を要求する. そして受け取った発言候補をランキング付けし議論支援システムに投稿を行う.

マネージャーエージェントのアルゴリズムを Algorithm 1 に示す. マネージャーエージェントは引数を 2 つ持ち, テーマ id, 議論支援システムの url によってファシリテーションを行う議論を指定する.

まずエージェントは議論支援システムの API を利用するためのアクセス権を入手する (1 行目). アクセス権を入手

#### Algorithm 1 マネージャーエージェント

**Input:** *theme\_id, system\_url*

**Parameter:** threshold *th*, confidence of each model *cm*

**Output:** None

- 1: get discussion log *d* from *theme\_id, system\_url*
- 2: request discussion structure *x* from Discussion Structure Extracting Agent by *d*
- 3: extract new posts *ns* from *d*.
- 4: **if**  $\text{count}(d) > th$  **then**
- 5:    $c \leftarrow [req\_nodes]$
- 6:   request suggested posts *sp* from Posts Generation Agent by *x, c*
- 7:   **for each** *p* of *sp* **do**
- 8:      $p.score \leftarrow p.confidence * cm(p.srcInfo)$
- 9:   **end for**
- 10:    $i \leftarrow \text{argmax}(sp.score)$
- 11:   posts *sp(i)* to system by *theme\_id, system\_url*
- 12: **end if**
- 13: **if** *d* equals null **then**
- 14:    $c \leftarrow [start\_discussion]$
- 15:   request suggested posts *sp* from Posts Generation Agent by *x, c, ns*
- 16:   select *p* from *sp*
- 17:   posts *p* to system by *theme\_id, system\_url*
- 18: **end if**

することで, 引数にて指定された議論テーマに対してファシリテートすることが可能となる. 次に議論構造抽出エージェントにリクエストを行なっている (2 行目). そして投稿ログから新規投稿の id を検出する (3 行目). 自動ファシリテータが返信文を作成する際は, ここで検出した新規投稿に対して返信する.

新規投稿の数が閾値よりも多かった場合, ファシリテータは返信を行う (4 行目). 閾値が 3 であれば, 参加者による投稿が 3 つ溜まる毎にファシリテータが 1 つ返信を返す. マネージャーエージェントは議論構造の情報, カテゴリー, および新規投稿の Id を発言生成エージェントに送る (5-6 行目). カテゴリーとは返信文の種類のこと, 投稿の意味ごとにカテゴリーが異なる. カテゴリーを指定することで,

マネージャーエージェントは任意の文章を得ることが可能となる。そしてマネージャーエージェントはスコアを計算する (7-8 行目)。発言候補には以下の情報が含まれる。

**confidence:** 投稿を生成するために用いた議論構造の信頼度。

**srcInfo:** 投稿を生成するために用いたモデルの情報。

各投稿は議論構造に基づいて生成される。ここでスコアは各投稿の信頼度、モデルの情報によって算出される。異なるモデルから信頼度の計算を行なった場合、信頼度の尺度が異なる。そのためモデルの情報を追加し confidence の調整を行った。

スコアの算出方法は以下の式を用いた。

$$score(i) = confidence(i) * cm(srcInfo(i)) \quad (4)$$

$confidence(i)$ ,  $srcInfo(i)$ , および  $cm$  は、それぞれ  $i$  番目の発言の生成に用いた議論構造の信頼度,  $i$  番目の confidence に用いたモデルの情報, および srcInfo 毎に設定したパラメータの値である。

マネージャーエージェントはスコアが最大の投稿を選び (10 行目), そして議論支援システムに投稿する (11 行目)。

”議論を開始しましょう”といった, 最初の投稿のような投稿は例外的な処理である。例外的な処理は条件を満たす場合に, カテゴリーを指定し投稿文を生成することで, 例外的な処理に対応している (13-17 行目)。

## 4.2 発言生成エージェント

発言生成エージェントの役割は, マネージャーエージェントから送られてきた, 議論構造の推定結果, カテゴリー, および新規投稿の投稿 id を元にテンプレートから条件に合う発言候補を生成することである。発言候補のフォーマットを統一することで, 複数体の発言生成エージェントを想定することができる。そのため異なるアプローチでファンリテータを行うシステムも取り込むことが可能である。

テンプレートには target, source, category, および text といった情報が含まれ, それぞれ返信先のノードの種類, 返信先に付くノードの種類, 発言の種類, および投稿文を意味する。事前に用意したテンプレートをもとに, 適切な投稿を行うことが可能となる。

発言生成エージェントのアルゴリズムを Algorithm 2 に示す。カテゴリーが投稿への返信である場合, 新規投稿に対応するノードの情報が抽出される (2-4 行目)。発言生成エージェントはノードの情報をもとに投稿を生成する (5-12 行目)。次のステップでは, どのノードにも返せるような汎用的な発言生成を行う (13-16 行目)。モデルを使用していない発言のため, confidence は target 毎に設定した値を設定し, srcInfo には none を設定する。ノード抽出やリンク抽出の confidence が低い場合にここで生成した文章が優先される。

## Algorithm 2 発言生成エージェント

**Input:** discussion structure  $x$ , category  $c$ , new posts  $ns$

**Parameter:**  $thresholds$

**Output:**  $sp$

```

1: initialize  $sp$  with empty list
2: if  $c$  equals "req_nodes" then
3:   for each  $n$  of  $ns$  do
4:     extract nodes from  $x$  matching  $n$ 
5:     for each node of nodes do
6:       target ← node.type
7:       source ← None
8:       generate posts  $ps$  matching the category with target,
       source,  $c$  from the template
9:       select  $p$  from  $ps$ 
10:      confidence ← node.confidence
11:      srcInfo ← name of the node classification model
12:      append { $p$ , confidence, srcInfo} to  $sp$ 
13:      generate posts  $sp$  matching the category with  $c$  from
       the template
14:      select  $p$  from  $ps$ 
15:      confidence ← thresholds(node.type)
16:      append { $p$ , confidence, srcInfo = "None"} to  $sp$ 
17:      extract links from  $x$  matching node
18:      for each link of links do
19:        target ← link.targetnode.type
20:        source ← link.source.node.type
21:        generate posts  $ps$  matching the category with
        target, source,  $c$  from the template
22:        select  $p$  from  $ps$ 
23:        confidence ← link.confidence
24:        srcInfo ← name of the link extraction model
25:        append { $p$ , confidence, srcInfo} to  $sp$ 
26:      end for
27:    end for
28:  end for
29: end if
30: if  $c$  equals "start_discussion" then
31:   generate posts  $ps$  matching the category with  $c$  from the
   template
32:   select  $p$  from  $ps$ 
33:   append { $p$ , confidence = 1.0} to  $sp$ 
34: end if
35: return  $sp$ 

```

次の処理として, 取り出したリンクを用いて文章を生成している (17-25 行目)。target に link 先のノード, source には link 元のノードが入る。上述の例と同様にテンプレートからリンク抽出を用いた発言候補を生成する。confidence にはリンク抽出のモデルの確信度を使い, srcInfo にはリンク抽出のモデルであることを示す link の値を代入している。最後に, 発言生成エージェントは発言候補を返却して処理を終了する (Line 35)。

## 4.3 議論構造抽出エージェント

議論構造抽出エージェントは, マネージャーエージェントに議論構造の推定結果を渡す役割を持つ。

アルゴリズムを Algorithm 3 に示す。議論構造抽出エー

**Algorithm 3** 議論構造抽出エージェント

**Input:** discussion log  $d$ ,  $theme\_id$ ,  $system\_url$   
**Parameter:** None  
**Output:** discussion structure  $x$

- 1:  $n \leftarrow$  number of posts in  $d$
- 2: make  $key$  from  $theme\_id$ ,  $system\_url$
- 3: **if**  $key$  in  $old\_cnt$  **then**
- 4:   **if**  $n$  equals  $old\_cnt[key]$  **then**
- 5:      $x \leftarrow old\_x[key]$
- 6:   **return**  $x$
- 7:   **end if**
- 8: **end if**
- 9: request node result  $nodes$  from node classification model by  $d$
- 10: request discussion structure  $x$  from link extraction model by  $nodes$
- 11:  $old\_cnt[key] \leftarrow n$
- 12:  $old\_x[key] \leftarrow x$
- 13: **return**  $x$

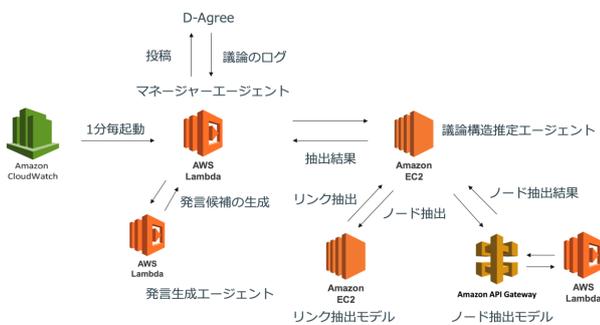


図 2 システムアーキテクチャ

エージェントはログから投稿の数を算出する (1 行目)。そしてテーマ id とシステム url から一意に定まる値を生成する (2 行目)。生成した値をもとに、過去に投稿に対して議論構造の推定を行っておりキャッシュが残っているか (3 行目)、投稿数に変化はあるかを判定する (4 行目)。キャッシュが残っており、投稿数に変化がないのであれば、過去の抽出結果を返却する (5-6 行目)。上記の条件を満たさない場合、エージェントはノード抽出モデルと、リンク抽出モデルに議論構造抽出のリクエストを行う (9-10 行目)。最後にエージェントは結果を返却し、抽出結果をキャッシュに残して処理を終了する (11-13 行目)。

4.4 アーキテクチャ

提案手法のアーキテクチャを図 2 に示す。エージェント間の通信には Web API を用いた。提案手法はクラウドサービス上に実装をしており、参加者の数によりスケールアップをすることが容易に可能である。また本提案手法ではエージェント毎に役割を分けた形式で実装してあるため、モデルやエージェントを入れ替えることが容易である。

5. 実験

5.1 前準備

表 1 データセット

class	num
Issue	169
Idea	864
Pros	451
Cons	440

表 2 Doc2Vec のパラメータ設定

Parameter	Value
vector_size	400
window	20
min_count	20
workers	11
alpha (Learning rate)	0.05
min_alpha	0.025
epochs	1000

D-Agree 上で収集された議論をもとに、Doc2Vec を用いてノード抽出のモデルを生成する。表 1 にデータセット中の各クラスの数、Doc2Vec のパラメータを表 2 に示す。構築したモデルを K-分割交差検証 [22] によって精度を測定した。表 3 に実験の結果を示す。Issue, Idea は比較的高い値を示しており、正しく分類することができていた。提案する自動ファシリテータは議論を正しく誘導するように発言を行った。特に Issue, Idea の認識が重要であるため、自動ファシリテータはある程度正しく議論構造の認識が可能と考えられる。

5.2 実験 1: 公開の議論に対するファシリテーション

名古屋市次期総合計画の中間案策定に向け、市民から意見を募るために一般に公開された形式で議論を行った。開催期間は、2018 年 11 月 1 日から 2018 年 12 月 7 日であり、ページビュー: 15199, 訪問人数: 798, 登録者数: 157, 総投稿数が 452 の議論規模となった。参加者はファシリテータが人間であるか、本提案エージェントであるかが分かる状態で実験を行なった。

表 4 にテーマ毎の投稿数と参加者の投稿数を示す。人間ファシリテータ (Human FA) がテーマ 1, 2 のファシリテーションを行い、自動ファシリテータ (Auto FA) がテーマ 3, 4 のファシリテーション、そしてテーマ 5 では人間と自動ファシリテータの両方がファシリテーションを行なった。

提案する自動ファシリテータエージェントが動作した例を図 3 に示す。投稿者は「防災、減災に対する正しい知識を身につける」という投稿を行っている。投稿に対して自動ファシリテータは課題と判断し、課題を解決するためのアイデアを求めている。結果として「起こりうる災害の種類を具体的にあげる」、「国内で起きた災害の知見をまとめる」、および「ゲーミフィケーションでバーチャルに訓練」というアイデアを引き出すことに成功している。しかし一方で、現状のファシリテータは返信先の投稿を要約して発言することはないため、人間がファシリテータを行

表 3 ノード抽出結果 (k=5)

	Doc2vec + Cos 類似度			
	Issue	Idea	Pros	Cons
precision	0.672	0.706	0.436	0.447
recall	0.858	0.702	0.299	0.540
f_value	0.751	0.703	0.353	0.486

表 4 テーマ毎の投稿数

Theme	All	Participants
Theme 1 (Human FA)	81	38
Theme 2 (Human FA)	56	35
Theme 3 (Auto FA)	88	64
Theme 4 (Auto FA)	70	52
Theme 5 (Auto and Human FA)	137	99
Sum	432	288

表 5 ファシリテーションの結果

Human FA		
全体の平均投稿数		156.8
ファシリテータの平均投稿数		25.2
ファシリテーションにかかった時間		399.9[s]
Issue の総計		28
Idea の総計		374
Pros の総計		144
Cons の総計		139
N/A の総計		91
Auto FA		
全体の平均投稿数		165.6
ファシリテータの平均投稿数		34
ファシリテーションにかかった時間		53.9[s]
Issue の総計		32
Idea の総計		344
Pros の総計		138
Cons の総計		146
N/A の総計		110



図 3 自動ファシリテータエージェントの一例

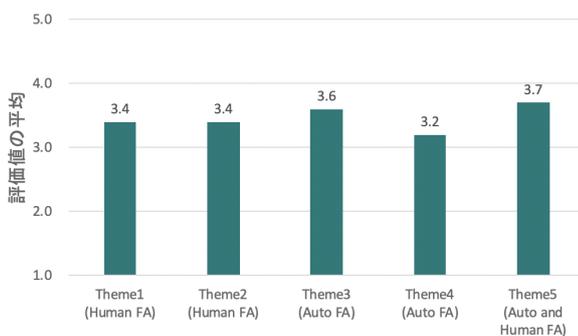


図 4 議論に満足することはできましたか? という質問に対するアンケート結果 (N=20)

うより柔軟に対応できてはいない。また次に「ゲーミフィケーションでバーチャルに訓練」というアイデアを正しく認識し、アイデアのデメリットを求める発言を生成している。結果として「ネットリテラシーが高くないと難しい」、「地域の特徴を訓練に反映させるには費用が高い」という意見の引き出しに成功している。

本実験終了後アンケートを行った結果を図 4 示す。アンケートの人数は 20 人であり、質問の内容は「議論に満足す

表 6 Human Facilitator

種類	要素数	参加者が行った投稿の種類				
		Issue	Idea	Pros	Cons	N/A
Response	32	0	9	7	6	3
Issue	4	0	4	1	3	0
Idea	46	0	52	2	1	1
Pros	8	0	1	6	0	0
Cons	12	0	2	2	15	0
Pros or Cons	5	0	0	2	2	0
合計	107	0	68	20	27	4

ることは出来ましたか?」という内容で、評価は 5 段階評価である。図 4 から本研究で提案する自動ファシリテータは人間のファシリテータと同等の値を獲得していることがわかる。以上の結果から本研究で提案した自動ファシリテータエージェントは有効に働いていたと考えている。

### 5.3 実験 2: 人間のファシリテータとの比較

提案する自動ファシリテータと人間のファシリテータを比較するために、同条件で議論支援の実験を行なった。参加者はファシリテータが人間であるか自動であるかが分からない状態で議論を行った。人間のファシリテータと自動ファシリテータがそれぞれ 5 テーマに対して、ファシリテーションを行い、議論時間は 45 分、または 60 分に設定した。

参加者の投稿中に含まれる IBIS 構造の要素数について表 5 に示す。要素数はファシリテータ別に示しており、人間のファシリテータと自動ファシリテータの場合で各要素の個数は同等である。従って本研究で提案するファシリテータが、正しく参加者の議論を認識し適切な投稿が行えたと考えられる。

表 6 と表 7 はそれぞれ人間のファシリテータと自動ファ

表 7 Automated Facilitator

Auto FA		参加者が行った投稿の種類				
種類	要素数	Issue	Idea	Pros	Cons	N/A
Response	35	0	6	6	7	3
Issue	3	0	0	0	2	1
Idea	77	0	65	0	4	8
Pros	13	0	1	7	1	2
Cons	35	1	1	1	21	2
Pros or Cons	3	0	0	0	0	0
合計	166	1	73	14	35	16

シリテータが参加者から引き出した要素の数を示す。提案する自動ファシリテータは人間と比較しても同等の意見を引き出すことに成功している。しかし一方で自動ファシリテータが議論構造の認識を誤り発言すると、参加者が誤った投稿に反応するため、全体としてN/Aの投稿が増えるといった傾向が見られた。

## 6. 結論

本研究では、建設的な議論支援を行う自動ファシリテータの手法を提案した。提案する自動ファシリテータはIBIS構造に基づき、議論を構造化した上で投稿を行う。自動ファシリテータは大規模議論に向けて、クラウドサービス上に実装を行なった。評価実験を通して提案手法が有効に働き、議論支援が可能であることを確認した。今後の課題としてIBIS構造以外の要素への対応、スコア算出方法の改良、議論構造の推定精度の向上が挙げられる。

謝辞 研究内容は、JST CREST「エージェント技術に基づく大規模合意形成支援システムの創成：代表伊藤孝行」(グラント番号JPMJCR15E1)に支援を受けている研究の一部である。

## 参考文献

- [1] Akihisa Sengoku, Takayuki Ito, Kazumasa Takahashi, Shun Shiramatsu, Takanori Ito, Eizo Hideshima, Katsuhide Fujita. plus.33emminus.07emDiscussion tree for managing large-scale internet-based discussions. plus.33emminus.07em*Collective Intelligence*, 2016, 2016.
- [2] Kazumasa Takahashi, Takayuki Ito, Takanori Ito, Eizo Hideshima, Shun Shiramatsu, Akihisa Sengoku, Katsuhide Fujita. plus.33emminus.07emIncentive mechanism based on quality of opinion for large-scale discussion support. plus.33emminus.07em*Collective Intelligence*, 2016:16, 2016.
- [3] Takayuki Ito, Yuma Imi, Motoki Sato, Takanori Ito, Eizo Hideshima. plus.33emminus.07emIncentive mechanism for managing large-scale internet-based discussions on collagree. plus.33emminus.07em*Collective Intelligence*, 2015, 2015.
- [4] Ronald E Rice, Adrian Shepherd, William H Dutton, James E Katz. plus.33emminus.07emSocial interaction and the internet. plus.33emminus.07em*Oxford handbook of Internet psychology*. 2007.
- [5] Alfred P Rovai. plus.33emminus.07emFacilitating online discussions effectively. plus.33emminus.07em*The Internet and Higher Education*, 10(1):77–88, 2007.
- [6] Daniel V Eastmond. plus.33emminus.07emEffective facilitation of computer conferencing. plus.33emminus.07em*Continuing Higher Education Review*, 56(1):23–34, 1992.
- [7] Takayuki Ito. plus.33emminus.07emTowards agent-based large-scale decision support system: The effect of facilitator. plus.33emminus.07em*Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.
- [8] Charlotte N Gunawardena, Frank J Zittle. plus.33emminus.07emSocial presence as a predictor of satisfaction within a computer-mediated conferencing environment. plus.33emminus.07em*American journal of distance education*, 11(3):8–26, 1997.
- [9] Steven R Aragon. plus.33emminus.07emCreating social presence in online environments. plus.33emminus.07em*New directions for adult and continuing education*, 2003(100):57–68, 2003.
- [10] Zachary Wong, Milam Aiken. plus.33emminus.07emAutomated facilitation of electronic meetings. plus.33emminus.07em*Information and Management*, 41(2):125 – 134, 2003.
- [11] A Adla, P Zarate, J-L Soubie. plus.33emminus.07emA proposal of toolkit for gds facilitators. plus.33emminus.07em*Group Decision and Negotiation*, 20(1):57–77, 2011.
- [12] Wen Gu, Ahmed Moustafa, Takayuki Ito, Minejie Zhang, Chunsheng Yang. plus.33emminus.07emA case-based reasoning approach for automated facilitation in online discussion systems. plus.33emminus.07emstrony 15–17, Pattaya, Thailand, 2018. The Thirteenth International Conference on Knowledge, Information and Creativity Support Systems (KICSS-2018).
- [13] Yuto Ikeda, Shun Shiramatsu. plus.33emminus.07emGenerating questions asked by facilitator agents using preceding context in web-based discussion. plus.33emminus.07em*2017 IEEE International Conference on Agents (ICA)*, strony 127–132. IEEE, 2017.
- [14] Douglas Noble, Horst WJ Rittel. plus.33emminus.07emIssue-based information systems for design. plus.33emminus.07em1988.
- [15] Trevor JM Bench-Capon, Paul E Dunne. plus.33emminus.07emArgumentation in artificial intelligence. plus.33emminus.07em*Artificial intelligence*, 171(10-15):619–641, 2007.
- [16] Sanjay Modgil, Henry Prakken. plus.33emminus.07emA general account of argumentation with preferences. plus.33emminus.07em*Artificial Intelligence*, 195:361–397, 2013.
- [17] Takayuki Ito, Yuma Imi, Takanori Ito, Eizo Hideshima. plus.33emminus.07emCollagree: A facilitator-mediated large-scale consensus support system. plus.33emminus.07em*Collective Intelligence 2014*, 2014.
- [18] Quoc V. Le, Tomas Mikolov. plus.33emminus.07emDistributed representations of sentences and documents. plus.33emminus.07em*CoRR*, abs/1405.4053, 2014.
- [19] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. plus.33emminus.07emEnriching word vectors with subword information. plus.33emminus.07em*Transactions of the Associa-*

- tion for Computational Linguistics*, 5:135–146, 2017.
- [20] Mike Schuster, Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
- [21] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.
- [22] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society. Series B (Methodological)*, 36:111–147, 1974.