

ファシリテーター支援のための分散表現を用いた話題変化判定

芳野魁†

†名古屋工業大学情報工学科

伊藤 孝行††

††名古屋工業大学大学院情報工学専攻

1 はじめに

近年、Web 上での大規模な議論活動が活発になっている。Web 上での議論には”5ちゃんねる (旧2ちゃんねる)” や”Twitter” といったシステムが一般に広く使われている。しかし、既存の情報共有システムでは、議論の整理や収束を行うことが困難である。故に、Web 上での大規模合意形成の実現を目的として、伊藤孝行研究室は過去に大規模意見集約システム COLLAGREE[1]を開発した。COLLAGREE は自由に意見を投稿や返信することができる掲示板のような議論プラットフォームをベースにしている。

他のシステムで議論の整理や収束を行うことが困難であった原因として議論の管理を行う者がいないことが挙げられる。前述の問題に対応するため COLLAGREE ではファシリテーターと呼ばれる人物による適切な議論プロセスの進行を行っている。しかし、長時間に渡って大人数での議論の動向をマネジメントし続けるのは困難である。COLLAGREE 上で使用されている議論支援システムは「(1) 投稿支援システム」と「(2) 議論可視化システム」の2つに大別できる。投稿支援システムはユーザーが投稿をする際に何らかの補助やリアクションを行う。例として高橋ら [2] が作成したポイント機能や伊美ら [3] が作成したファシリテーションフレームズ簡易投稿機能が挙げられる。現行の投稿支援システムでは選択肢の提示に留まっているため、作業量を減らすことには繋がるとは言い難い。一方、議論可視化システムはユーザーに対してスレッドとは異なる議論の見方を提供する。例として仙石ら [4] が作成した議論ツリーやキーワード抽出機能が挙げられる。しかし、現行の議論可視化システムでは議論を見やすくすることに重点が置かれているため、議論の把握の助けにはなるが画面に向き合う時間を減らすとは言えない。むしろ、作業内容によっては作業量を増やす可能性がある。従って、現行の支援機能ではファシリテーターの作業量の減少には繋がっていない。

また、分散表現は単語を実数ベクトルで表現する技術である。分散表現は自然言語処理の分野において多くの研究で使われている。分散表現を使用することで従来の手法よりも単語の意味や関連性を考慮した処理が可能になる。分散表現は単語の共起頻度に基づいて類似度を計算しているため、同じ文脈で出現することの多い対義語に対応できない点が弱点とされる。しかし、議論中の話題においては対義語かどうかは関係なく同じテーマ (話題) に沿っているかが重要である。分散表現における類似度計算は議論における話題の繋がりと似ているため、分散表現の対義語にできないというデ

メリットを無視して話題変化判定ができると考えられる。故に、分散表現は話題変化判定と相性が良いと考えた。

以上の背景を踏まえて、本研究ではファシリテーターを支援するために分散表現を用いて自動的な話題変化判定を目指す。本研究の目的はファシリテーターの代わりに議論中の話題変化を判定することである。

本論文の構成を以下に示す。2章では、COLLAGREE の概要やユーザーの議論を支援するファシリテーターに関する説明、及び関連研究について述べると共に本研究との違いに言及する。更に、本研究の重要な要素である重み付けと分散表現についても詳しく述べる。次に、3章では話題変化判定システムの全体モデル説明を行い、4章では分散表現を用いた発言内容の類似度計算について説明する。そして、5章では話題の変化判定の評価実験について説明する。最後に6章で本論文のまとめと考察を示す。

2 関連研究

2.1 COLLAGREE

概要

COLLAGREE は各ユーザーが自由なタイミングで意見を投稿や返信することのできる掲示板のような議論プラットフォームをベースにしている。図1にCOLLAGREE での実際の議論画面を示す。COLLAGREE のような議論掲示板は基本的には1つの議論テーマに対して関連するテーマを扱った複数のスレッドから構成される。スレッドとはある特定の話題・論点に関する1つのまとまりを指す。図2にCOLLAGREE でのスレッドの例を示す。図2では四角形が発言を表し、発言を結ぶ線は返信関係を表す。図2では発言0を親意見としてスレッドが立てられている。そして、他のユーザーが子意見として親意見に返信し、更に子意見に対して孫意見が存在している。

ファシリテーター

COLLAGREE ではファシリテーターと呼ばれる人物が議論のマネジメントを行っている。ファシリテーターは議論そのものには参加せずあくまで中立的な立場から活動の支援を行うため、自分の意見を述べたり自ら意思決定をすることはない。ファシリテーターの基本的な役割として”議論の内容の整理”、”議論の脱線防止”、”意見の促し”等が挙げられる。ファシリテーターが様々な論点に対する発言を促すことによって議論が発散するため、十分な議論が行われる。ファシリテーターの存在や手腕が合意形成に強く影響を与えることやWeb 上での議論においてファシリテーターが有用であることが伊藤ら [5] や伊美ら [3] によって示されている。

A Topic Change Judgment Method based on Distributed Representation for Facilitator Support

Kai Yoshino Takayuki Ito

†Department of Computer Science, Nagoya Institute of Technology

††School of Techno-Business Administration, Graduate School of Engineering Nagoya Institute of Technology



図 1: COLLAGREE の議論画面

2.2 議論支援

小谷ら [6] は好意的発言影響度を取り入れた議論支援システムを開発した。システムは議論中の発言の意図や内容に加えて、発言に対するリアクション (同意, 非同意, 意見) などから議論進行をモニタリングしている。そして、モニタリングの結果を基にして議論の活性化や深化に対して参加者が果たしている役割を”好意的発言影響度”として定量化して表示する。

本研究と小谷らの研究は両研究とも発言が議論に与える影響を扱っている点で関連している。しかし、小谷らは一般参加者や学習者の議論活性化及び収束に向けた支援を目的としている。一方で本研究はファシリテーターに対する支援を目的としている点で異なる。

2.3 話題遷移検出

話題の遷移に基づいた文章の分割が主に人間によるテキスト全体の内容把握を容易にすることや複数のテキストに対する自動分類や検索精度向上を主な目的として研究されている。

別所ら [7] は単語の共起頻度行列を特異値分解で次元を削減して作成した単語の概念ベクトルと TextTiling [8] を用いてトピック変化点を検出し、連結された新聞記事を元の形になるように分割をする実験を行っている。図 3 に TextTiling を用いて作成されたグラフとグラフの値に基づく分割地点を示す。

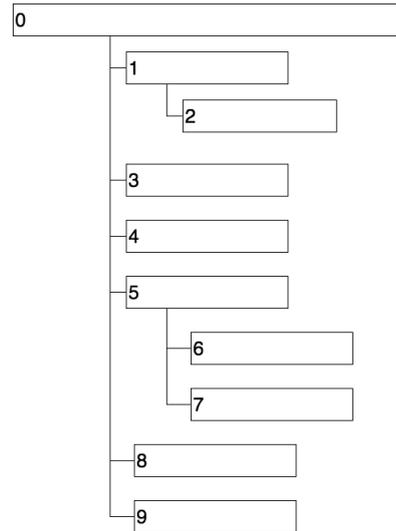


図 2: COLLAGREE のスレッドの例

本研究と別所らの研究は両研究とも文字列間の類似度を話題に基いて計算している点で関連している。しかし、図 3 に示すように、TextTiling はグラフ全体の中からブロック (複数の単語を連結したもの) 間類似度の小さい地点から分割を行うため、分割する地点よりも未来の情報を使っておりリアルタイムな議論での動作には適さない。本研究は分割する地点よりも未来の情報を使うことなく話題変化を判定する。

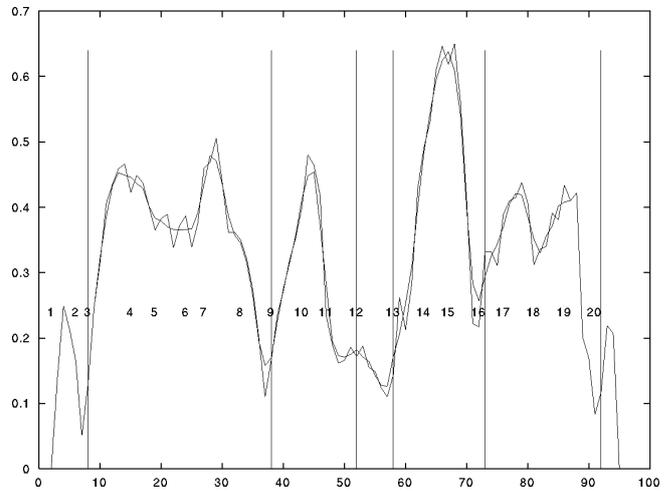


図 3: TextTiling のグラフ

2.4 分散表現を使用した話題関連研究

1章で述べたように分散表現は単語の関連性を導出できることから分散表現を話題関連に用いる研究が行われている。

中野ら [9] は分散表現を用いて雑談対話システムでのシステム側の応答生成を行っている。図 4 に応答生成の構造を示す。中野らのシステムではユーザ発言を形態素解析して検出された単語を単語分散表現による類似語検索から得られた結果と入れ替えることでシステ

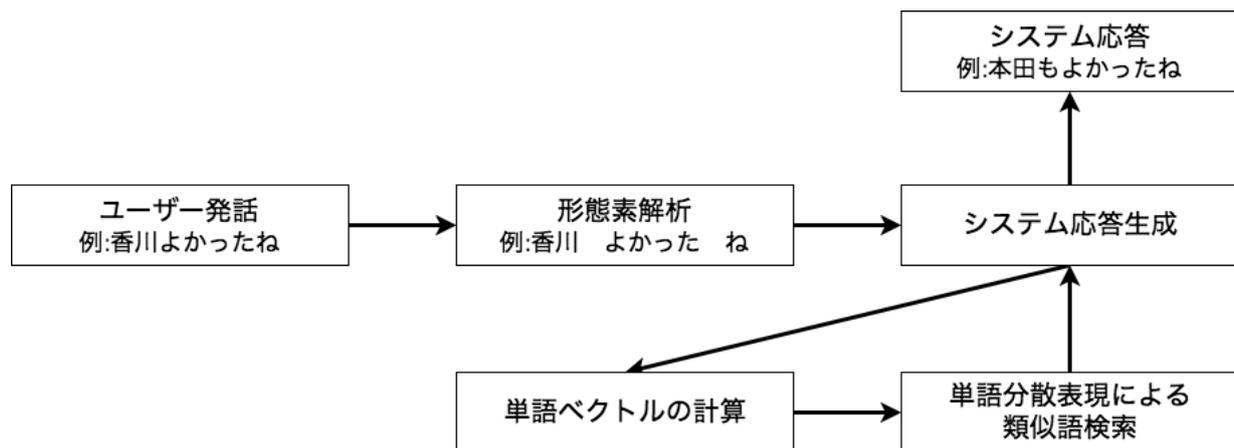


図 4: 中野らによる応答生成の構造

ム応答の生成を行っている。本研究と中野らの研究は両研究とも分散表現を話題展開または変化に対して関連している。しかし、中野らが類似単語を導出するために分散表現を使用しているのに対して、本研究は単語集合間の類似度を計算するために分散表現を使用している点で異なる。

3 話題変化判定システム

3.1 システム動作の流れ

Algorithm1 を用いて話題変化判定システムの動作の流れについて説明する。本システムでは話題の変化は発言 R と過去の発言の集合である PG の比較によって判定する。まず最初に、新しく投稿された発言 R で

Algorithm 1 話題変化判定システムの流れ

```

1: Input : 発言 R
2: Output : 通知判定 Notify
3: PG = 過去の発言の集合;
4: procedure TOPICCHANGE(R)
5:   SG = {};
6:   update(R)
7:   for Each pastR ∈ PG do
8:     sim = similarity(R,pastR)
9:     if sim > threshold then
10:      SG.append(pastR)
11:   Notify = False
12:   if SG == {} then
13:     Notify = True
14:   return Notify
    
```

使われている単語の情報を 4 章で説明する重み付けで用いるために単語の出現回数等の更新を行う (6 行目)。そして、 PG に含まれる過去の発言と R の類似度を計算し (8 行目)、類似度が閾値を超えていた場合、同じ話題である発言集合 SG に比較した 2 発言の話題が同じであるとして登録する (9~10 行目)。全ての比較が終了した後 SG が空集合である、すなわち発言 R と同じ話題である発言がない場合、話題を変化させる発言であると判定する。

3.2 類似度計算

発言間の類似度計算は次の 3 段階で行われる。

① 前処理

本研究では前処理として、発言の内容文を対象に②で行われる発言内容の類似度計算の精度を上昇させるためにストップワードの除去や単語の重み付けを行う。前処理の詳細については 4 章で述べる。

② 発言内容の類似度計算

①で行われた前処理の情報や分散表現を用いて発言内容文の類似度計算を行う。文章間の類似度計算の手法については 4 章で詳しく述べる。

③ 総合類似度計算

上記の②で計算された発言の文章間の類似度に発言間の時間差と返信関係を組み合わせることで総合類似度を求める。時間差評価値は式 (1) に示すように、発言間の投稿時間差を最大時間差で割り 1 から引くことで時間差評価値 $tValue$ は 0 から 1 の範囲となる。2 発言間の時間差が小さいほど関連が強いとみなされて 1 に近くなる。

$$tValue = 1 - \frac{new.created - old.created}{maxTime} \quad (1)$$

$maxTime$ は最大時間差を表す数値である。 $maxTime$ は基本的に議論の制限時間を用いる。 $x.created$ は発言 x が投稿された時間を表す。総合類似度に時間差評価値を導入して時間的に近いものほど総合類似度を上昇させることで、議論が基本的に少し前に発言に関連して進行されることが多いという点を考慮した。具体的には式 (2) のように計算される。

$$total = tValue * tWeight + sim * (1 - tWeight) \quad (2)$$

$time$, sim はそれぞれ時間差評価値と発言内容の類似度を表す。 $tWeight$ は時間差評価値の重みを示す。 $tWeight$ は 0 から 1 の値を取る。また、2 発言が同じスレッドに属していた場合は何らかの関連があると考えられる。

故に、2 発言が同じスレッドに属していた場合は式 (3) に示すように時間差評価値を無視し、発言内容の類似度に補正值 α を加えたものを総合類似度とした。

$$total = sim + \alpha \quad (3)$$

4 発言内容の類似度計算

本章ではストップワード除去と重み付けによる前処理及び発言内容の意味的類似度を計算する手法を説明する。

4.1 ストップワード除去

分散表現による類似度計算で精度を上昇させるためには発言内容から余分な単語を取り除き重要な単語を抽出するか極めて短く要約することが重要である。本研究では形態素解析エンジン MeCab[10] による形態素解析の結果、出力フォーマットは次に示す形式となった。

表層形 \t 品詞, 品詞細分類 1, 品詞細分類 2, 品詞細分類 3, 活用型, 活用形, 原形, 読み, 発音

上記のフォーマットの元、次の条件を満たす単語を除外している。

1. 品詞細分類に「数」を含む
2. 「読み」, 「発音」が不明である
3. 品詞が「助詞」, 「助動詞」, 「記号」, 「連体詞」のどれかである。
4. 1 文字のひらがなである
5. 品詞細分類に「接尾」または「非自立」を含む

4.2 重み付け

提案手法では okapiBM25[11] と LexRank[12] の 2 種類の重み付け手法を統合して発言文章 *remark* 中の単語に対して重み付けを行う。アルゴリズムを **Algorithm2** に示す。本研究において固有名詞は文章の中で重要な役割を果たす可能性が大きいと考えたため、固有名詞の単語重みを 4 倍にしている。(12 行目)。word を含む全文章の重みの合計を求める (14 ~ 16 行目)。そして、okapiBM25 による単語重みを掛け合わせたものを word の統合重みとしている (17 行目)。単語重みに単語を含む文章の重みを掛け合わせることで感動詞のような単語を含む文章そのものは重要でないが、重要性が高いと判断されてしまう単語が選ばれる可能性を下げている。

4.3 類似度計算

本研究では計算された単語重みの上位 n 個までの単語を発言文章 *remark* において重要度の高い単語であるとして抽出する。それぞれの発言から抽出された単語を分散表現を用いて単語をベクトルに変換した後、平均ベクトルの Cosine 類似度を取ることで発言文章間の類似度としている。本研究では分散表現として fastText[13] を用いる。

Algorithm 2 統合重みの計算アルゴリズム

```

1: Input : remark 発言内容の文字列
2: Output : combinedWeight remark 中の単語と重みを対応付けた連想配列
3: Array sentList; ▷ 以前に重み付けを行った最大  $n$  個前までの文章のリスト
4: procedure CALCCOMBINEDWEIGHT(remark)
5:   bm25Weight = calcBM25Weight(remark)▷ 単語と重みの連想配列
6:   for Each sent ∈ remark do ▷ remark を句点、改行コードで分割する
7:     sentList.append(sent)
8:   lexWeight = calcLexRank(sentList)
9:   for Each word ∈ bm25Weight.keys() do
10:    wordWeight = bm25Weight[word]
11:    if word is 固有名詞 then
12:      wordWeight *=4
13:    sentWeight = 0
14:    for Each sent ∈ remark do
15:      if word in sent then
16:        sentWeight += lexWeight[sent]
17:    combinedWeight[word] = wordWeight*sentWeight
18:   return combinedWeight

```

5 評価実験

本章では、COLLAGREE で行われた議論のデータを対象にした提案手法の評価実験について述べる。

5.1 実験データ

評価実験では COLLAGREE 上で行われた議論時間 90 分の 2~3 名による複数の議論データを使用する。結果として提案手法が分散表現と単語抽出を用いていることで良い精度を出せるか確認する。議論データに対し、次に述べる基準で学生にアノテーションを行ってもらった。本研究で用いた基準は筒井 [14] が会話内容を分析する上で立てた話題を区切る基準を参考にした。

1. それまで話題となっていた対象や事態とは異なる、新しい対象や事態への言及する発言
2. 既に言及された対象や事態の異なる側面への言及する発言
3. 議論のフェーズを移行させる (可能性の高い) 発言
4. ファシリテーターによる議論をコントロールするような発言

アノテーション担当者が基準を満たすと判断した発言に True のタグを、満たさずと思われない発言に False のタグを付けてもらった。そして、True のタグが過半数付けられた発言を話題変化発言とした。

5.2 実験設定

比較手法として以下に述べる 3 つを用いた。

比較手法 1 : 常に話題変化発言として判定する

比較手法 2 : 発言文章を TF-IDF[15][16] による単語の重みを用いてベクトル化したものを fastText の代わりに用いる

比較手法 3 : 発言文章を LDA[17] を用いてトピックベクトル化したものを fastText の代わりに用いる

提案手法 1 ではパラメーターは次の通りに設定した。前処理にて用いる okapiBM25 のパラメーターは $k_1 = 2, b = 0.75$ とし, LexRank では 50 個前までの文を用いた。また, 重み付けを用いて文章から抽出する単語の数は 5 個とした。fastText は次元数を 100 次元とし, 学習データには wikipedia ダンプデータを用いた。wikipedia ダンプデータを用いた理由として, 分散表現を作るにはデータ量が多い方が良く, 汎用的なデータで分散表現を作り議論のテーマに対して特定の学習を行うことなく話題変化判定を行いたいと考えたことが挙げられる。式 (2) の $tWeight$ は 0.5 とし, 総合類似度の閾値は 0.8 とした。

また, 単語抽出が精度上昇に貢献していることを確認するために提案手法 2 では単語抽出を行わずに文章中の全単語の平均ベクトルで内積を取る。

評価指標として適合率 (Precision), 再現率 (Recall), F 値 (F-measure) の 3 種類の指標を用いる。

5.3 実験結果

実験結果として各手法の F 値を表 1 に示す。各手法の再現率と適合率及び 2 つの値の差を表 2 に示す。

手法	F-measure
比較手法 1 (常に話題変化発言)	0.404
比較手法 2 (TF-IDF ベクトル)	0.487
比較手法 3 (LDA ベクトル)	0.385
提案手法 1 (単語抽出あり)	0.515
提案手法 2 (単語抽出なし)	0.086

表 1: 実験結果 1

また, 各手法において話題変化発言と判定した発言の割合を示す P-SUM と話題変化発言でないと判定した発言の割合を示す N-SUM を表 3 に示す。

5.4 考察

実験結果から, 以下の考察①, 及び考察②が言える。

考察① 提案手法 1 は比較手法よりも総合的に性能が高い

表 1 が示すように, 提案手法 1 は他のどの手法よりも高い F 値を出している。一方で, 表 2 が示すように, 提案手法では再現率と適合率の間の差が 0.037 で最も小さくなっている。以上から, 適合率と再現率のバランスが最も良かったことが他の手法よりも高い F 値に繋がったと思われる。

手法	平均評価指標		
	Precision	Recall	Difference
比較手法 1 (常に話題変化発言)	0.257	1	0.743
比較手法 2 (TF-IDF ベクトル)	0.489	0.558	0.069
比較手法 3 (LDA ベクトル)	0.573	0.317	0.256
提案手法 1 (単語抽出あり)	0.515	0.552	0.037
提案手法 2 (単語抽出なし)	0.833	0.046	0.787

表 2: 実験結果 2

手法	平均割合	
	P-SUM	N-SUM
比較手法 1 (常に話題変化発言)	1.0	0
比較手法 2 (TF-IDF ベクトル)	0.83896	0.16103
比較手法 3 (LDA ベクトル)	0.255	0.745
提案手法 1 (単語抽出あり)	0.58776	0.41223
提案手法 2 (単語抽出なし)	0.02737	0.97262

表 3: 実験結果 3

考察② 単語抽出を行うことで性能が上昇する

表 1 が示すように, 提案手法 1 は提案手法 2 よりも高い F 値を出している。一方で, 表 2 が示すように提案手法 2 は非常に高い適合率を示すと同時に, 非常に低い再現率を示している。提案手法 1 と提案手法 2 の間の違いは単語抽出を行うかどうかだけなので結果に影響を与えたのは明らかに単語抽出である。単語抽出が結果に影響を与えた理由として, 単語抽出によって平均を取る単語ベクトルの数が減ったことが考えられる。表 3 が示すように, 提案手法 2 では P-SUM 数が他の手法に比べて非常に少ない上, 多くの発言が過去の発言のどれかと高い類似度を示していることが伺える。

すなわち, 平均を取る単語ベクトルの数が多くなることで単語に関係なく結果的に平均ベクトルが全て似たようなものとなったため, 発言の平均ベクトルの差が小さくなってしまったと想定できる。

6 おわりに

本章では本研究のまとめ及び今後の課題について述べる。

6.1 まとめ

本研究では分散表現を用いてファシリテーターの代わりに自動的に話題の変化を判定することを目標とする。話題変化の判定は, 新しく投稿された発言と過去

に投稿された発言との類似度を計算してどれか類似しているものがあるかどうかで判定する。発言の類似度は発言文中に現れる単語の類似度と見なすことができる。類似度は、発言中の特徴的な単語を分散表現に変換して内積計算によって求める。発言文から単語を選ぶ際には自動要約を用いる。発言文から重要な単語だけを取り出すことで類似度の計算における精度を高めることが可能となる。具体的な提案手法は、LexRankによる評価値の高い文の中のokapiBM25の評価値の高い単語を抜き出す。そして、選ばれた単語を分散表現に変換したものの内積を取ることで類似度を求める。

上記の提案手法に対し、比較手法として常に話題変化発言として判定する手法とTF-IDFによる単語ベクトルを使う手法、そしてLDAによる話題ベクトルを使う手法を用いて、議論中の話題変化判定の評価実験を行った。評価実験によって、提案手法を用いることで比較手法よりも高い性能で話題の変化を判定できることと単語抽出によって精度が上昇していることを確認した。

6.2 今後の課題

本研究における今後の課題としては、以下の点が挙げられる。

課題1 COLLAGREEでの実装及び実証実験

本研究ではCOLLAGREEで事前に行われた議論のデータを対象に評価実験を行なったが、COLLAGREEでの実装は行っていない。実装にあたり、実際の議論に適用した修正が必要となることが予想される。また、本研究では実際の議論におけるファシリテーターによる評価が行われていない。検討している実証実験ではファシリテーターに重要な発言を話題変化判定システムが判定できているかについてと、話題の変化判定が有る無しによってファシリテーションの違いが現れるのかについての2項目について明らかにしたい。

課題2 類似度計算の改善等による適合率の上昇

本研究では提案手法が比較手法よりも良い性能を示すことができたが、提案手法の適合率は高いとは言えない。本研究では抽出的要約を使用していたが、発言文によって重要な単語の適切数は異なる。抽出される単語の数が多すぎれば重要でないが意味的に類似している単語によって類似度が上昇してしまうことが起こりうる。一方で、単語の数が少なすぎれば意味的に重要な単語の数が足りず適切な類似度にならないことが起こりうる。抽出される単語に依存せず類似度計算を行うために発言文全てを類似度計算に用いる手法が必要となる。具体的にはWord Mover's Distance[18]による類似度計算やDoc2Vec[19]を用いた発言文全体のベクトル化等を検討している。

課題3 話題の繋がりの可視化及び相互作用化

本研究ではファシリテーターの負担を軽減することを目的として話題変化判定を行った。しかし、話題の繋がりが分かるというのはファシリテーターだけでなく、一般の参加者にとっても議論把握の支援になり得る。具体的には仙石らが開発した議論ツリーのような形式での議論支援を検討している。

参考文献

- [1] 伊藤孝行, et al. "多人数ワークショップのための意見集約支援システム collagree の試作と評価実験". 日本経営工学会論文誌, pp. 83-108, 2015.
- [2] 高橋一将, 伊藤孝行, 伊藤孝紀, 秀島栄三, 白松俊, 藤田桂英, et al. 大規模意見集約支援のための意見内容と投稿タイミングに基づくインセンティブ機構. 第78回全国大会講演論文集, Vol. 2016, No. 1, pp. 409-410, 2016.
- [3] 裕麻伊美, 孝行伊藤, 孝紀伊藤, 栄三秀島. オンラインファシリテーション支援機構に基づく大規模意見集約システム collagree一名古屋市次期総合計画のための市民議論に向けた社会実装. 情報処理学会論文誌, Vol. 56, No. 10, pp. 1996-2010, oct 2015.
- [4] 仙石晃久, 伊藤孝行, 藤田桂英, 白松俊, 伊藤孝紀, 秀島栄三. "web 上での大規模議論における議論ツリーによる意見集約支援". 研究報告知能システム, pp. 1-8, 2016.
- [5] 田中 恵伊藤 孝紀, 伊藤孝行, 秀島栄三. "ファシリテータに着目した合意形成支援システムの検証と評価". デザイン学研究, pp. 67-76, 2015.
- [6] 小谷哲郎, 関一也, 松居辰則, 岡本敏雄. 好意的発言影響度を取り入れた議論支援システムの開発. 人工知能学会論文誌, Vol. 19, No. 2, pp. 95-104, 2004.
- [7] 別所克人ほか. 単語の概念ベクトルを用いたテキストセグメンテーション. 情報処理学会論文誌, Vol. 42, No. 11, pp. 2650-2662, 2001.
- [8] Marti A Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, Vol. 23, No. 1, pp. 33-64, 1997.
- [9] 中野哲寛, 荒木雅弘. 雑談対話システムにおける単語分散表現を用いた話題展開手法. 言語処理学会第21回年次大会発表論文集, pp. 269-272, 2015.
- [10] Taku Kudo. Mecab: Yet another part-of-speech and morphological analyzer. <http://taku910.github.io/mecab/>.
- [11] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, Vol. 109, p. 109, 1995.
- [12] Gunes Erkan and Dragomir R. Radev. "lexrank: Graph-based lexical centrality as salience in text summarization". *Journal of Artificial Intelligence Research*, pp. 457-479, 2004.
- [13] Bojanowski and Piotr et al. "enriching word vectors with subword information". *arXiv preprint arXiv:1607.04606*, 2016.
- [14] 筒井佐代. 雑談の構造分析. くろしお出版, 2012.
- [15] Hans Peter. Luhn. "a statistical approach to mechanized encoding and searching of literary information". *IBM Journal of research and development*, pp. 309-317, 1957.
- [16] Karen Sparck Jones. "a statistical interpretation of termspecificity and its application in retrieval". *Journal of Documentation*, Vol. 28, pp. 11-21, 1972.
- [17] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, Vol. 3, No. Jan, pp. 993-1022, 2003.
- [18] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pp. 957-966, 2015.
- [19] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188-1196, 2014.