

第3章

話題変化判定システム

3.1 序言

本章では話題変化判定システムの概要について説明する。

以下に本章の構成を示す。まず3.2節でシステム動作を流れ図から説明する。次に3.3節ではシステムの詳細を説明すると共にシステムで扱う発言データの形式や発言間の類似度計算について述べる。最後に、3.4節で本章のまとめを示す。

3.2 システム動作の流れ

擬似コードを **Algorithm1** に示し、図示したものを図3.1に示す。

Algorithm1 を用いてシステムの動作の流れについて説明する。本システムは話題変化を判定するため、話題の変化させる発言を以下の手順で把握する。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。発言 R が投稿された時に過去の発言集合である PG 内に発言 R と同様の話題がないと判定された場合、同じ話題である発言集合 SG が空集合となり、話題を変化させる

発言であるとする。また、同じ発言の判定は、発言 R が投稿された時、過去の発言の集合である PG と発言 R の類似度を計算する。以上で計算された類似の値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 つの話題が同じであるとして登録する。上記の過去の発言との比較作業は **Algorithm1** の 5~9 行目に相当し、作業を繰り返して全ての発言との計算が終了した後、発言 R に返信先がなく、 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に話題を変化させる発言であると判定して通知を行う。

Algorithm 1 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:   for Each  $pastR \in PG$  do
7:      $sim = \text{similarity}(R, pastR)$ 
8:     if  $sim > \text{threshold}$  then
9:        $SG.append(pastR)$ 
10:   $Notify = \text{False}$ 
11:  if  $SG == \{\}$  then
12:     $Notify = \text{True}$ 
13:  return  $Notify$ 

```

3.3 システム詳細

3.3.1 発言データ

本システムで扱う”発言データ”は単なる文字列ではなく、タイムスタンプ等の他のデータを持つリスト形式のデータである。表 3.1 にデータの一部を示す。また、以下に本研究で使用する発言データの要素について説明する。

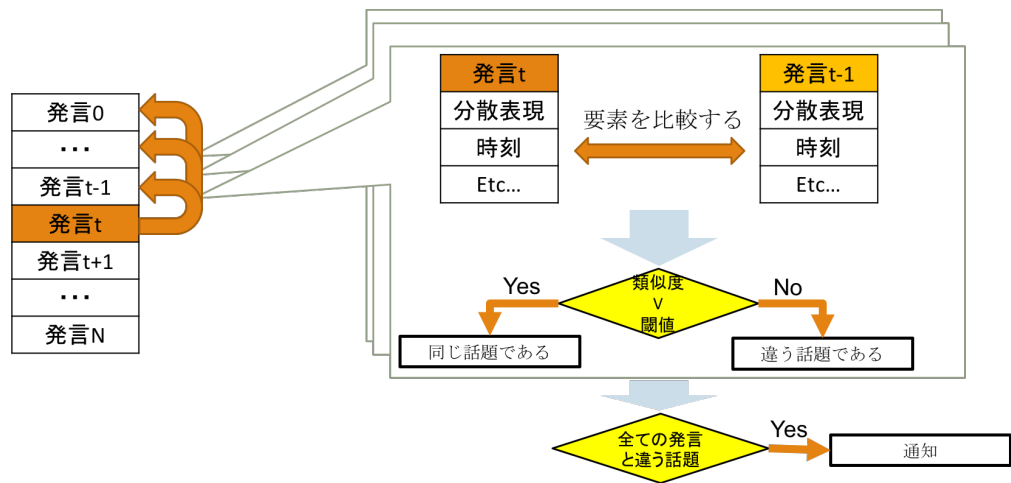


図 3.1: システムの流れ

① id

発言データを識別するための番号であり，全て整数値で表される．

② title

スレッドのタイトル名を表す文字列で，発言がスレッドの先頭でない限りは NULL となる．

③ body

発言の内容を表す文字列である．

④ parent-id

発言の返信先，すなわち親発言の id を表す番号で，親発言がある場合は親発言の id と同じ番号になり，ない場合は NULL となる．

⑤ created-at

発言が投稿された時間を示すタイムスタンプである。

id	title	body	parent-id	created-at	(以降は省略する)
18	オンラインでの議論に関する実験	参加者の皆様が集まるまでお待ち下さい。	NULL	2017/03/14 16:28:00	(以降は省略する)

表 3.1: 発言データ

3.3.2 発言間の類似度計算

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、発言データ中の title と body、すなわち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(役立たないことから処理対象外とされる単語)の除去や単語の重み付けを行う。また、title が NULL でない場合は title と body を改行コードで繋いで1つの文章とする。前処理の詳細については??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??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

新たに投稿された発言 *new* と *new* 以前の発言の 1 つである *old* 間の時間差を式 3.1 に基いて正規化された評価値として求める。

$$tValue = 1 - \frac{epoch(new.created) - epoch(old.created)}{maxTime} \quad (3.1)$$

ここで関数 *epoch* 及び定数 *maxTime*, *x.created* について説明する。 *epoch* は与えられたタイムスタンプをエポック秒に変換する。 *maxTime* は最大時間差を表し、基本的には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。時間差評価値は 0 から 1 の範囲で示され、2 発言間の時間差が小さいほど関連が強いとみなし、1 に近くなる。

返信距離

返信距離は発言 *new* と以前の発言 *old* 間の返信距離を Algorithm2 に基いて再帰的に求められる。

5 ～ 6 行目で示すように 2 発言間が返信関係になかった場合は 0 を返す。 2 発言間が返信関係にある場合は 7 ～ 8 行目、 9 ～ 12 行目で示すように発言の id が一致した場合は現在の返信距離を返し、一致しなかった場合は返信距離を 1 増やして *new* の親発言 *parent* と *old* の返信関係を再帰呼び出しで求め、返り値を返す。

Algorithm 2 返信距離

```
1: Input : 発言 new, 発言 old, 返信距離 dist ▷ 初期値 dist=1
2: Output : 返信距離 dist
3: PG = ID に対応づけられた過去の発言の集合;
4: procedure REPLYDIST(new, old, dist)
5:   if new.parent-id == NULL then
6:     return 0
7:   else if new.parent-id==old.id then
8:     return dist
9:   else
10:    parent = PG[new.parent-id]
11:    dist + = 1
12:    return REPLYDIST(parent, old, dist)
```

総合類似度

総合類似度は前述の発言内容の類似度，時間差評価値，返信距離によって求められる．返信距離が0である時，すなわち2発言が異なるスレッドに属している場合は類似度と時間差評価値から総合類似度を計算する．総合類似度に時間差評価値を導入した理由は，基本的に少し前に発言に関連して議論が進行されることが多いため，時間的に近いものほど評価を上昇させる仕組みを考慮したからである．時間差評価値を使用して時間的に近いものほど総合類似度が上昇するようにする．具体的には式 3.2 のように計算される．

$$totalSim = tValue * tWeight + sim * (1 - tWeight) \quad (3.2)$$

式 3.2 の変数 *sim*, *tValue* 定数 *tWeight* について説明する．*sim* は発言内容の類似度を表し，0 から 1 の値を取る．*tValue* は式 3.1 での *tValue* と同じもので時間差評価値を表す．*tWeight* は時間差評価値の総合類似度の計算における重要性を表し，0 から 1 の値を取る．また，返信距離が0でない，すなわち2発言が同じ

スレッドに属している場合は何らかの関連があると考えられることから，式 3.3 に示すように時間差評価値を無視して発言内容の類似度を直接，総合類似度として用いる．

$$totalSim = sim \quad (3.3)$$

3.4 結言

本章では話題変化判定システムの動作の流れやアルゴリズムについて説明し，扱うデータの形式や内容を示した．??章にて説明する発言内容の類似度の計算手法の詳細を除いて，発言間の類似度計算の手法について説明した．また，総合類似度は、時間的に近い議論ほど類似度が上昇するように時間差評価値と発言内容の類似度から求めることを示した．