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 節でシステム全体の動作の流れを示し, アルゴリズムについても説明を行う．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　システムの動作の流れ

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．発言 R が投稿された時，過去に投稿された発言と類似度の計算を行い類似度が閾値を超えていれば 2 つの発言が同じ話題であるとみなし，発言 R と同じ話題である発言の集合 SG に登録する．作業を繰り返し全ての発言との計算が終了した後，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

```
Input : 発言 R
Output : 通知判定 Notify
PG = 過去の発言の集合;
procedure TOPICCHANGE(R)
    SG = {};
    for Each pastR ∈ PG do
        sim = similarity(R,pastR)
        if sim > threshold then
            SG.append(pastR)
    Notify = False
    if SG == {} then
        Notify = True
    return Notify
```

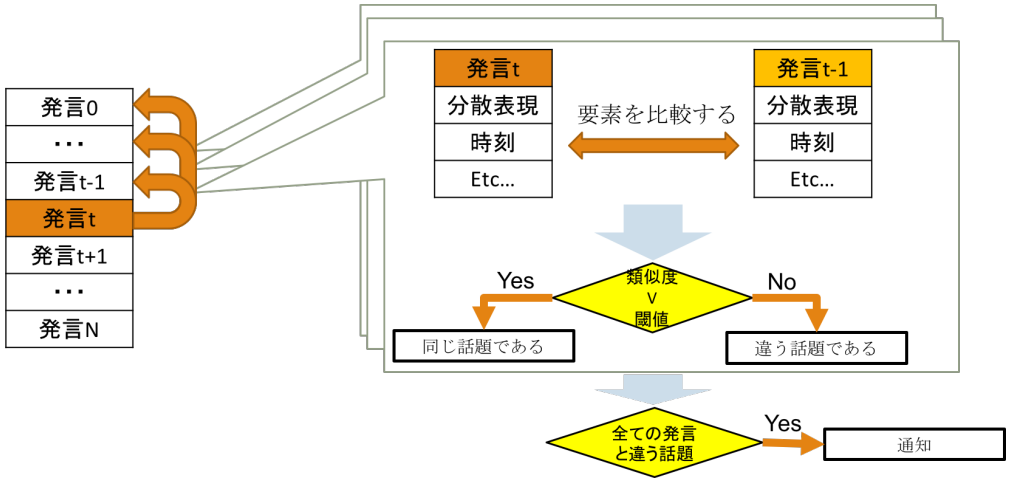

図 3.1: システムの流れ

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，以下で本研究で使用する発言データの要素について説明する．

① id

発言データを識別するための番号で，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．

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 crreated-at

発言が投稿された時間を示すタイムスタンプ．

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

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

発言 new と以前の発言 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．

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

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 2 発言間の時間差が小さいほど関連が強いとみなし，0 から 1 に近づく．

返信距離

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る．

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)
```

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．また，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．

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．類似度だけでなく時間差評価値を使用するのは，総合類似度だけで判断してしまうと議論の終盤になって発言数が多くなってきた時に新しく投稿された発言が多くの古い発言と類似していると判断されてしまうことがあり得るからである．議論は基本的に少し前の発言に関連して行われることが多いことから時間差評価値を使用して時間的に近し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．

$$tSim = tValue * tWeight + sim * (1 - tWeight) \tag{3.2}$$

ここで変数 sim，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．$tWeight$ 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，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．

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も説明した．また，時間的に近し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の２つを用いて総合類似度計算することを示した．