

電子メールを利用したコミュニケーションにおける 討議スレッド自動抽出法の実装と評価

山見 太郎† 村越 広享† 島津 明† 落水 浩一郎†

†北陸先端科学技術大学院大学 情報科学研究科
〒923-1292 石川県能美郡辰口町旭台 1-1

Email: {t-yamami,murakosi,shimazu,ochimizu}@jaist.ac.jp

あらまし

ソフトウェア開発などの共同作業における電子メールコミュニケーションを対象とした討議スレッド自動抽出法の改良と、討議スレッド自動抽出を行うシステム(討議スレッド抽出エンジン)の実装を行う。電子メールを利用したソフトウェア仕様レビュー作業の3事例を対象として、抽出エンジンによる討議スレッド抽出を行い、抽出結果に対する評価を行う。その結果、抽出エンジンおよび自動抽出法の有用性と改善点を明らかにした。

Development and Evaluation of Deliberation Threads Extraction Engine for Email Messages

Taro YAMAMI† Hiroyuki MURAKOSHI†
Akira SHIMAZU† Koichiro OCHIMIZU†

†School of Information Science,
Japan Advanced Institute of Science and Technology,
1-1, Asahidai, Nomi-gun, Ishikawa, 923-1292 Japan

Email: {t-yamami,murakosi,shimazu,ochimizu}@jaist.ac.jp

Abstract

In this paper, we focus on communication using email in a cooperative work such as software development. We improve the method of automatic extraction of deliberation threads from email messages and develop an automatic extraction engine which extracts them from email messages. Then we extract the deliberation threads of three examples of software specification reviews using this engine and analyze the cause of extracting wrong deliberation threads. From the results of analysis, we examine the usefulness of this engine and the method.

1 はじめに

1.1 背景と目的

電子メールを利用したコミュニケーションでは、対面の場合と同様の話者交代を行うことができないため、複数の話題を同一メール内に記述する傾向にあるといえる。そのため、複数の話題が並列に議論

されやすい。また、聞き手が自分の発話を理解しているかどうかを認識するための手がかりも少ないといえる。そのような状況では、対話者間の会話に関する共通理解の阻害と、認識の不一致の発生が予想される [1]。

共同ソフトウェア開発において、コミュニケーションにおける作業間認識の不一致は、ソフトウェ

ア最終成果物に悪影響を及ぼすことが考えられる。過去の発話を話題毎に分類、整理し、発話意図の聞き手の理解状況を確認でき、適切な返答を行なっているかといった討議の流れを参照することができれば、話題の終結に関する作業員間の認識の不一致を回避したり、同じ話題の議論の繰り返しが少なくなるといった効果があると思われる [2]。例えば、討議の道筋とその根拠などの情報を系統的に管理するシステムが開発されている [3][4]。

本研究では、討議の道筋を発話ペアを基本とした同一話題の連鎖(以降、討議ストリーム)として抽出する自動抽出法 [5][6] の実装をおこなう。以後、この実装物を抽出エンジンと呼ぶ。さらに、抽出エンジンを用いて自動抽出を行った結果を評価することにより、抽出エンジンおよび自動抽出法の評価を行う。

1.2 関連研究

近年、テキストを用いた非同期コミュニケーションのメッセージからの討議の構造の抽出、提示に関する研究が行われている。例えば、ネットワークニュースを対象に、参照関係を示す木構造と話題転換記事の推定を行うことによって、ユーザが指定した記事と関連する記事群を抽出して提示するシステム [7] や、電子メールを対象に、引用文を利用することにより討議のスレッドの抽出と視覚化を行うシステム [8][9] の開発が行われている。

2 討議ストリームとその言語的特徴

2.1 討議ストリーム

対面での会話の場合、話者交代ごとに一つの発話生成されると捕らえることができる。しかし、電子メールを利用したコミュニケーションの場合、対面での話者交代とは異なるため、発話を話者交代によって定義することができない。したがって、電子メールを利用したコミュニケーションにおける発話を定義する必要がある。

同一メール内に複数の話題が存在しうることが、電子メールを利用したコミュニケーションの特徴の一つである。そこで、発話を、同一メール内で同じ話題に言及している文章と定義する。ただし、発話を構成する文章間には、同一話題という整合性(コヒーレンス)が維持されるものとする。発話は、以下の三種類に分類することができるものとする。

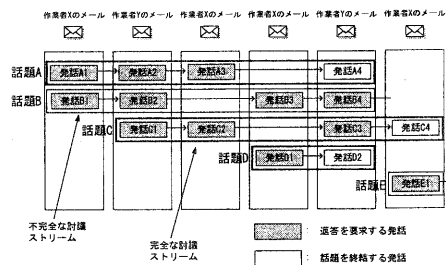


図 1: 討議ストリームによる討議の構造

- 相手からの返答を求めず、議論の進行に直接の影響を与えない発話(以降、宣言的な発話)
- 相手からの返答を要求する発話(以降、返答を要求する発話)
- 相手からの返答を求めず、話題を終結する発話(以降、話題を終結する発話)

討議ストリームを、同一話題に言及している1つ以上の返答を要求する発話と話題を終結する発話による連鎖と定義する。討議ストリームの最初の発話は、返答を要求する発話である。返答を要求する発話に続く発話は、返答を要求する発話か、話題を終結する発話となる。討議ストリームが話題を終結する発話で終結している場合、そのストリームを完全な討議ストリームと呼ぶ。反対に、話題を終結する発話で終結していない討議ストリームを、不完全な討議ストリームと呼ぶ。

以上の定義によって構成される討議の構造の例を図1に示す。図1では、メールを縦長の長方形で、メールに含まれる発話を小さい長方形で表現している。また討議ストリームを、水平のラインでリンクされている発話の列で表現している。

2.2 討議ストリームの言語的特徴

2.2.1 電子メールボディ部の特徴

電子メールはヘッダ部とボディ部から構成される。討議ストリームを構成する発話はボディ部に含まれる。電子メールを利用したコミュニケーションでは、対面での対話と異なり、会話の一時停止などを利用した話題転換の指示ができない。その代わりに、文章中に意図的に空行や改段落を挿入することによって話題転換の指示を行なっているものと考えられる。しかし、空行や改段落で分割される文章が発話単位であることが多いものの、分割されているものの一つの発話であるものもある。発話の最低単位は一文

次のふたつについて
以下の2点です
挙げます
列挙します

表 1: 新たな話題の並列展開を示唆する表現例

変更に関して
第1章のことですが
4の予約キャンセルについてですが

表 3: 新たな話題の開始を示唆する表現例

であり、一文中に空行や改段落があっても一文が発話であることが多い。あるメールに対する返答を含むメールのメッセージ部では、引用文が利用されているという特徴がある。引用文とは、各行の先頭に“>”, “>”といった引用符を伴ってあらわれる文章であり、以前の議論であらわれた文章を引用している文章のことである。引用文は、引用文の直後にあられる発話の先行発話を示すことが多い。また、メッセージの最初は、自分の名前や身分を示す宣言的な発話であることが多く、メッセージの最後は、一般に「署名」と呼ばれるものであることが多い。署名は、自分の名前や身分や連絡先などをまとめたもので、討議と直接の関係はない。署名が存在する場合は、署名の直前に“---”, “-----”といった記号だけからなる行(セパレータ)を伴うことが多い。

2.2.2 文章表現にあらわれる特徴

新たな話題の並列展開を示唆する表現 相手からの返答を求めない発話の中には、新たに議論される複数の話題を明示的に示しているものがある。たとえば、「挙げておきます」のような表現は、続く空行や段落で分割される文章を、それぞれ異なった話題の最初の発話とするための手がかりになることが予想される。そこで、新たな話題の並列展開を示唆する表現に着目する。その表現例を表1に示す。

話題の転換を示唆する表現 会話における話題の転換では、それまでの話題を終始、あるいは停止し、新たな議論の開始を示唆する表現が用いられることがある [10][11]。同一メール中の隣接した発話間においても、発話の先頭文中に、話題転換を示唆する文副詞、接続詞があることがある。その例を、表2に示す。また、明示的に新たな話題の開始を示唆することによって、話題の転換を示唆することもある。

指示語	名詞的	このうち、この部分、この場合
	文副詞的	これは、これに
接続語		具体的には、例えば

表 4: 結束性を示す表現例

語尾	例
か	思いませんか
かな	私の提案で良いかな
ましょう	移りましょう
でしょう	どうでしょう
よ	時間がずれているよ
ね	決める必要がありますね

表 6: 返答要求を示唆する表現例

その例を、表3に示す。

結束性を示唆する表現 メール中の空行や改段落によって分割された文章同士でも、同一の文章で、一つの発話となっている場合がある。これは、発話者が文章の長さや読みやすさを考慮し、空行や改段落を用いて一つの話題の文章を分割しているためであると考えられる。語、文、発話の線状的な連鎖の言語形式による「つながり」を示す、結束性という概念がある [12]。この結束性は、同一話題の文章を見つげるための手がかりとなる。結束性を示唆する表現には、指示語による表現と接続語による表現があり、どちらも後半の文章の先頭文中に現れる。それぞれの例を表4に示す。

話題の終結を示唆する表現 話題を終結する発話の場合、文章の先頭文の文頭あるいは文末に、相手の発話を理解し、同意を示す表現が含まれていたり、先頭文の文頭に感嘆を示す語が含まれる傾向にある。その上、先頭文以外の文末に、後述する返答の要求を示唆する表現が含まれない傾向にある。表5に話題の終結を示唆する表現例を示す。

返答の要求を示唆する表現 返答を要求する発話には、相手への質問や提案などを示す語、つまり相手への返答の要求を示唆する表現が含まれていることが多い。表6にその表現例を示す。

対を構成する表現 発話ペアを構成する先行発話の最後の文末と後続発話の先頭文には、相手への問いかけと、それに対応した同意を示す表現の対が存在することがある。これらの対は、発話ペアを見つけ

	話題転換を示唆する文副詞, 接続詞
順序を示す表現	あと, まず最初に, 以下の, 次は, 次に
追加的表現	それから, さらに, また
話題変化を示す表現	そう言えば, ところで

表 2: 話題転換を示唆する文副詞・接続詞の例

明示的な肯定表現	指示語を用いた同意表現	感嘆を示す表現
はい	そうだね	確かに
了解です	そうだよね	そうか
わかりました	そうですね	なるほど
賛成です	そう思います	ふんふん
同意します	そのとおりですね	まったくですな

表 5: 話題の終結を示唆する表現例

先行発話 (問いかけ)	後続発話 (応答)
バグとしますか	バグですね
矛盾していると思いませんか	思います

表 7: 対を構成する表現例

るための手がかりになるだけでなく, 後続発話が話題を終結する発話であるかを決定する手がかりにもなると思われる。その表現例を表 7 に示す。

2.2.3 言語的特徴のまとめ

以上の討議ストリームの言語的特徴を, 討議ストリームを抽出するための言語的手がかりとして表 8 にまとめた。討議ストリーム自動抽出法はこれらの言語的手がかりを用いて抽出を行う手法である。

3 自動抽出法の実装

文献 [6] 等で定義された自動抽出法には, 引用部分や改段落などの文章構造の判定, 名前や身分を判定する文章の判定, 言語的手がかりの発見方法など, 具体的なアルゴリズムが明記されていない部分がある。実装を行う際には, これらを明確にする必要がある。そこで, 以下に変更した抽出手順を示す。

手順 1 発話候補となる文章集合の抽出

手順 2 発話の抽出と分類

手順 3 先行発話の決定

3.1 発話候補となる文章集合の抽出

手順 1 では, ボディ部を分割し, 発話候補となる文章の集合を抽出する。その際に用いる手がかりと分割手順の定義をおこなう。

3.1.1 分割の手がかり

発話候補となる文章への分割は, 電子メールボディ部に現れる特徴のうち, 空行と改段落, 引用符, セパレータ手がかりとして用いる。実装にあたって, それぞれを次のように定義した。

空行 間隔文字だけからなる行を空行とする。

改段落 行頭が間隔文字であり, 行中に間隔文字以外の文字を含む行があれば, その行から新しい段落が始まるとする。

引用符 次のパターンにマッチする文字列を引用符とする¹。

```
/^(?:.{0,9})[|>||>>][L \t]*+)$/
```

セパレータ 次のパターンにマッチする文字列を本文と署名を分けるセパレータとする。

```
/^[L \t]*?[-=+_~---__+*]+$/
```

3.1.2 分割手順

具体的なボディ部の分割手順は次の通りである。対象行とは処理を行う対象である行のことを示し, 前の対象行とは, 現在の対象行が対象行となったときの対象行を示し, 次の行とは, 対象行の直後の行を示す。

手順 1-1 ボディ部の最後から数行を対象に, セパレータの探索をおこなう。探索は上からボディ部末に向かって行い, セパレータが見つかり次第, そのセパレータからボディ部の最後までを発話の候補からはずす。探索が終わり次第, 対象行をボディ部の先頭行に変更する。

¹本稿で用いるパターン表記は JPerl5 における正規表現の表記法と同じである。

示唆する特徴	略号	存在位置	例
話題の並列展開	clue1	文章中	列挙します, 挙げます
話題の転換	clue2	先頭文の文頭	ところで, ~については
接続語による結束性	clue3	先頭文の文頭	具体的に, 例えば
指示語による結束性	clue4	先頭の文中	この場合, これは
話題の終結	clue5	先頭文の文頭/文末	わかりました, そうですね, 確かに
返答の要求	clue6	文末	か, かな
対となる表現	clue7	文末-文末	~と思いませんか-思います

表 8: 言語的手がかりの一覧

手順 1-2 対象行が引用符を伴っているならば, 引用符を伴わない行まで読み飛ばす. 対象行をその引用符を伴わない行に変更する.

手順 1-3 対象行が空行なら, 空行でない行まで読み飛ばす. その空行でない行を対象行として, 手順 1-2 に戻る.

手順 1-4 対象行が改段落であり, かつ前の対象行が空行または引用文でなければ, 対象行を新たな文章の始まりとする. 次の行を対象行として, 手順 1-2 に戻る.

手順 1-5 対象行は発話の候補となる文章を構成する行である. もし, 前の対象行が引用文でも空行でもなければ, 対象行は前の対象行を含む文章を構成する文章である. そうでなければ, 新たな文章を構成する行である. 次の行を対象行として, 手順 1-2 に戻る.

3.2 複数文章から発話の抽出

手順 1 の結果, 発話の候補となる文章の集合が得られる. 次に, 得られた文章集合から発話を抽出する. 文献 [6] 等では引用文が現れる前後で異なる手続きを定義しているが, 共通部分が多い. そこで, 引用文の前後で処理を区別せず, 共通化することにした. 具体的な手順を以下に示す. ただし, 対象文章とは処理を行う対象である文章のことを示し, 続く文章とは, 対象文章の直後の文章のことを示す.

手順 2-1 続く文章が存在しない場合, 対象文章を返答を要求する発話として抽出する. もし, 対象文章の直後が引用文であり, その引用文の直後に文章がある場合, その文章を対象文章として, 手順 2-1 に戻る. そうでなければ, 発話の抽出終了.

手順 2-2 対象文章中に clue1 があるならば, 対象文章を宣言的な発話として抽出する. さらに, 対象文章から次の引用文までの文章群を, それぞ

れ返答を要求する発話として抽出する. もし, 次の引用文の直後に文章があるならば, その文章を対象文章として手順 2-1 に戻る.

手順 2-3 対象文章に続く文章がある場合, 手順 2-3a から手順 2-3d を行う.

手順 2-3a 続く文章の文頭に clue2 があるならば, 対象文章を返答を要求する発話として抽出する.

手順 2-3b 続く文章の文頭に clue3 があるか, 続く文章の先頭文中に clue4 があるならば, 対象文章と続く文書を結合し, 手順 2-1 に戻る.

手順 2-3c 対象文章の文末に clue6 がなければ, 対象文章と続く文章を結合し, 手順 2-1 に戻る.

手順 2-3d 手順 2-2a から手順 2-2c までの条件に該当しないならば, 対象文章を返答を要求する発話として抽出する. 引き続き, 続く文章を対象文章として, 手順 2-1 に戻る.

3.3 後続発話の分類

手順 2-3 までにより, 発話が抽出される. 次に, 各引用文の直後の発話を後続発話として, 返答を要求する発話か話題を終結する発話かのどちらかに再分類する. 具体的な手順は次の通りである.

手順 2-4a 対象の発話中に clue5 があり, 対象の発話を構成する各文章の文末に clue6 がなければ, 対象の発話を話題を終結する発話とする.

手順 2-4b 対象の発話中に clue5 がなくとも, 発話のペアを構成する発話間に clue7 があり, 対象の発話を構成する各文章の文末に clue6 がなければ, 対象の発話を話題を終結する発話とする.

手順 2-4c 以上に該当しない場合, 対象の発話を返答を要求する発話とする.

この手順では, clue5, clue6, clue7 を利用して分類を行うことになっているが, 今回は clue7 の実装を実現していない.

3.4 品詞情報を付加した言語的手がかり 辞書の利用

文章中に存在する言語的手がかりを正しく発見することができるかどうか、この手法による抽出結果を左右する。しかし、実際にはどのように発見すれば良いのかが明確になっていない。

言語的手がかりの発見を単純な文字列のパターンマッチで行う場合、次のような問題が生じる。

言語的手がかりとして品詞情報も必要な例 同一話題の文章を捉える特徴として、結束性を示す指示語を言語的手がかりとして用いるが、単純な文字列のパターンマッチでは、それが本当に指示語であるかを認識することは不可能である。

- おこのみですね。
- この話はなかったことにしませんか？

具体的な手がかりを与えることが難しい例 宣言的な発話を認識するための言語的手がかりとして、メールの送り手の名前や身分を宣言する文章を用いているが、単純な文字列のパターンマッチでそれらを判別しようとする場合、名前や所属などの情報の準備が必要となる。しかし、手間などの面から得策ではない。

これらの問題を解決するために、言語的手がかり辞書に形態素情報を付加し、手がかり検索の前処理として、対象文書の形態素解析を行う。

3.5 先行発話の決定手順

引用文を手がかりに、引用文直後の発話の先行発話を決定する。引用文を構成する各行の先頭文字列を手がかりに、対象のメール以前に届いたメールを到着順において近いものから順に検索し、同じ文字列を含む発話を先行発話とする(手続3)。

3.6 処理対象

自動抽出法は、対象とするメール集合の保存形式に依存しないが、今回は、MH形式で保存されたメールを対象とする。そこで、抽出エンジンの処理対象に次の3つの前提をおく。

- 対象とするメール集合は、すべて1つのMHフォルダの中に保存されている。また、そのフォルダには対象外のメールは存在しない。
- 対象となるメールは、すべてRFC822準拠である。
- メールボディ部のコードセットはISO-2022-JPである。

	事例1	事例2	事例3
メール数	30	75	46
発話数(正解/合計)	37/47	193/259	84/111
スレッド数(正解/合計)	12/22	29/90	23/37

表 9: 抽出結果

抽出エンジンは、これらの前提の元、MHフォルダディレクトリとメール番号によって指定されたメールに対して上記の処理を行うものとする。

3.7 実装環境

今回は、以下に挙げる理由により、JPerl5を用いて実装を行った。

- 漢字を含む文字列に対するパターンマッチを容易に記述できる言語である
- RFC822 準拠のメールと MH フォルダを取り扱うためのモジュールが存在している (Mail::Internet, Mail::Folder::Emaul)
- 日本語文字コードの変換を行うためのモジュールが存在している (Jcode)
- 形態素解析器に対するインターフェースとなるモジュールが存在している (Text::Chasen)

手がかり語の検索の際に用いる形態素解析器には茶筌[13]を用いる。今回の抽出エンジンは、以上の言語処理系、モジュール、形態素解析器が動作する環境であれば動作する。

4 評価

4.1 評価方法

電子メールを利用したソフトウェア仕様書のレビュー作業の3事例を対象に、抽出エンジンによる討議スレッドの抽出を行う。抽出結果を分析、抽出間違いの生じる原因を明らかにし、それらを分類をすることによって、抽出エンジンと自動抽出法に対する有用性の評価を行う。

4.2 抽出結果

各事例はそれぞれ30通(事例1)、75通(事例2)、46通(事例3)のメールからなる。各事例において抽出した発話とスレッドの抽出した数と正解の数を、表9に示す。なお、次のような発話とスレッドを不正解とした。

- 複数の話題を含む発話
- 発話を構成する文章が明らかに不完全な発話
- 話題の異なる発話を含む討議スレッド
- 不正解である発話を含む討議スレッド

Clue1	Clue2	Clue3	Clue4	Clue5	Clue6
6	13	3	5	22	10

表 10: 言語的手がかり辞書登録語彙数

4.3 抽出間違い発生の原因

4.3.1 言語的手がかり辞書の語彙不足

自動抽出法において言語的手がかりの正確な発見が重要であることは3.4節で述べたが、実際に言語的手がかりの発見を行う手続き2での手がかりの見落としは、発話とスレッド、両方の抽出精度の低下に直結する。抽出エンジンで用いた言語的手がかりの語彙数を表10に示す。現状でも、全事例において発話抽出の正答率は50%を越えているが、より多くの語彙を登録することによって、より正確な発話の抽出を目指すことが可能かつ必要であると思われる。

4.3.2 アルゴリズムの簡略化による影響

抽出エンジンの開発において行った自動抽出法の簡略化が原因で正答率が落ちている事例が存在した。以下にその詳細を述べる。

均等インデントの無視 事例3において、メッセージを均等にインデントしている事例が見られた。この場合、抽出エンジンは各行を改段落と捉えてしまうため、結果として、1つの発話とすべき文章を複数の発話として抽出してしまうことがある。このような均等なインデントを行っている事例が事例6で多く見られ、問題も発生していた。

安易なセパレータの検知法 外部コンテキストからの引用を行う場合等において、引用部と本文を分けるために引用符を用いず、本文と署名を分かちセパレータと同様の表現を用いることがある。事例1と2において、1件ずつこの事例が見られた。この場合、手順1-1において誤解が生じ、発話を構成する文章を署名として認識してしまい、発話候補から除去してしまうといった問題が生じている。

引用元を示す付加情報の存在の無視 先に送られたメールからの引用の場合、引用部の直前に「xxxx <xxxx@aaa.bb.cc> さんは書きました:」のような引用元を示す情報を伴うことがある。事例2に多く存在した。今回の実装では、このような引用元を示す情報を利用していないので、独立した返答を要求する発話として抽出されることが多かった。事例2

の発話抽出の正答率が他の事例と比べて低い理由の一つである。

引用符の定義 「>」のような記号は、引用符以外にも用いられる。例を以下に示す。

(例1) 発話対象明示 ありがたいですね > 設計者?

(例2) 矢印の代替 人数増加 → もっと大きな部屋
現在の引用符の定義では、このような用途で用いた記号も引用符として捉えてしまうことがある。今回は、事例2に、2件存在した。

Clue1の発見後の手順 手順2-2において、Clue1の発見後の手続きを定義しているが、現在の定義では、Clue1の存在した文章の直後から次の引用文までに存在する発話候補の文章それぞれを、そのまま発話として抽出してしまう。そのために発話の誤分割が生じた事例が、事例2に2件存在した。

箇条書きに対する処理の省略 発話候補となる文章への分割は手順1で行っているが、行頭に「・」や「●」などの記号などを伴って現れる箇条書きには対応していない。特に、箇条書きの2行目以降をインデントしている場合は、2行目以降の各行を改段落と判定してしまうため、発話の誤分割が生じやすい。実際に問題が生じた事例が、事例1に1件、事例2に3件、事例3に1件存在した。

4.3.3 自動抽出法で対応できない例

4.3.2節で取り上げた事例と異なり、自動抽出法の精度が原因で正答率が低下している事例が存在した。

先行発話を「挟み込んだ」発話 自動抽出法では、先行発話を示す引用は後続発話となる文章の直前に現れるとしている。しかし、以下に示すように、後続発話となる文章中に先行発話が挟み込まれている事例が事例1に1件、事例2と事例3に各2件存在した。

(例) 先ほどのメールにあったように、

> これはユーザ向けの言わば外部仕様書?

この仕様書は、

結果として、後続発話の誤分割と、それに伴う発話ペアの誤構築が生じてしまう。

引用の中略 長い文章を引用する際、引用文中を中略し、中略を行ったことを「(略します。)」等の文によって示す事例が存在した。自動抽出法は、このような状況を想定していないので、中略を示す文が返答を要求する発話として抽出されてしまう。

先行発話の暗示 事例3で特に顕著だったのが、引用による先行発話の明示が行われない事例であった。先行発話は「削除の件は」とか「変更に関して」等の形で暗示されているものの、同一メール中に先行発話を具体的に示す手がかりは存在しなかった。

4.4 評価

表9が示すとおり、現時点での抽出エンジンによる抽出精度は高いとは言えない。さらに、抽出間違いの原因の分析の結果、様々な問題が明らかになった。しかし、これらの問題は実装と評価実験の結果から明らかになった問題である。しかも、自動抽出法自体の価値をなくしてしまうような問題は見受けられなく、すべて解決可能な問題であることがわかった。したがって、今回の実装はプロトタイプ実装として価値のあるものであり、自動抽出法も改良の余地はあるものの、十分その役割を果たしうるものであるといえる。

5 まとめと今後の課題

本研究では、自動抽出法の改良を行い、自動抽出法の実装(抽出エンジンの開発)を行った。抽出エンジンを用いてソフトウェア仕様レビュー作業の事例から討議スレッドを抽出し、抽出エンジンと自動抽出法を評価した。自動抽出法と抽出エンジンの精度を高めるための今後の課題を以下に示す。

- 手がかり語辞書の充実
- 討議ストリーム自動抽出法の改良
- 抽出結果の視覚化

参考文献

- [1] 村越広享, 落水浩一郎. 電子メールを利用した共同作業における会話のコヒーレンスと話題の完結度の関係. コンピュータソフトウェア, Vol. 15, No. 3, pp. 50-53, 1998.
- [2] 桑名栄二. ソフトウェア履歴利用の研究動向. 電子情報通信学会誌, Vol. 77, No. 5, pp. 531-538, May 1994.
- [3] 近野章二, 門脇千恵, 落水浩一郎. グループウェアベース「葉」を用いた電子会議内容の進捗状況と文書化の支援. 情報処理学会研究報告 96-SE-107, Vol. 107, No. 12, pp. 89-96, Jan. 1996.
- [4] Koichiro Ochimizu, Chie Kadowaki, and Masakazu Hori. Design of an Information Repository to Support Cooperative Works over a Computer Network. In *IPSJ International Symposium on Next-Generation of Information Technologies*, pp. 79-86, 1997.
- [5] Hiroyuki Murakoshi, Akira Shimazu, and Koichiro Ochimizu. Construction of Deliberation Structure in Email Communication. In *Pacific Association for Computational Linguistics(PACLING'99)*, pp. 16-28, Aug. 1999.
- [6] Hiroyuki Murakoshi, Akira Shimazu, and Koichiro Ochimizu. A Method of Computing Measure for Evaluating Conversational Coherency in Email Communication. In *International Workshop on Collaboration and Mobile Computing(IWCMC'99)*, pp. 66-73. IEEE Computer Society Press, Sep. 1999.
- [7] 井佐原均, 小作浩美, 内元清貴. 討論型ニュースグループを対象とする知的ニュースリーダーの開発. 情報処理学会研究報告, Vol. 97, No. 53, pp. 13-18, May 1997. 自然言語処理.
- [8] 松浦文崇, 高田眞吾, 中小路久美代. 計算機上におけるコミュニケーションの履歴表示に関する研究. 情報処理学会研究報告 99-GW-31, Vol. 99, No. 7, pp. 43-48, Jan. 1999. グループウェア.
- [9] 村上明子, 長尾確. ディスカッションマイニング: 構造化されたコミュニケーションによるトピックの検索と視覚化. 言語処理学会第6回年次大会発表論文集, pp. 451-454, Mar. 2000.
- [10] Rachel Reichman. Conversational Coherency. *Cognitive Science*, Vol. 2, No. 4, pp. 283-327, 1978.
- [11] 泉子・K・メイナード. 会話分析. くろしお出版, 1993.
- [12] M. A. K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman, 1976.
- [13] 松本裕治, 北内啓, 山下達雄, 平野善隆, 浅原正幸. 日本語形態素解析システム【茶筌】 version 2.0 使用説明書 第二版. NAIST Technical Report NAIST-IS-TR99012, 奈良先端科学技術大学院大学, 奈良県生駒市高山町 8916-5, December 1999.