

引用文を活用したメールの要約*

伊知地 宏[†] 倉部 淳[‡]

[†] ラムダ数学教育研究所

[‡] インタービジョン・レーザーフィッシュ

概要

我々は、話題が関連するメール本文の要約を行って、メールでの議論の推移を簡単に把握できるソフトウェアの開発をしている。本稿では、そのソフトウェアの開発に際して行ったメールの要約方法について報告する。我々のメールの要約方法では、要約文を読むだけでメールの内容を理解できることを目指しており、引用文と重要単語に着目して要約を行う。要約文作成には、文の意味解釈をするのではなく、引用文のうち最後の1文と引用直後の3文、および出現頻度が上位3位までの重要単語を含む文を採用するという方式を取っている。このアルゴリズムを実装したソフトウェアで、ある程度まで我々の目標を満たす要約を得られることがわかった。あわせて、要約文が短く出来ない問題点と辞書に起因する重要単語の選択の問題点があることも明らかになった。

1 はじめに

携帯電話の普及に伴い、電子メールを利用する人が爆発的に増えてきた。ソフトウェアの開発においても、メールをソフトウェア開発の道具として使うことが多くなり、メールの重要性がさらに増している。

例えば、Linuxの開発のように、メールを使ってソフトウェアの仕様や実装方法の検討、ソースコードのレビューを行うようなソフトウェア開発が増えてきている。特に、居場所の異なる数人でソフトウェアを開発する際には、メールによる議論は非常に有効な方法である。実際、著者達が同じ会社にいた時代には共同でソフトウェアの研究開発を行う機会が多かったが、それぞれが場所の異なる別々のオフィスにいたため、顔を会わせて打合せをすることはほとんどなく、打合せ、議論そしてソフトウェアのデバッグまでとソフトウェアの研究開発のほとんどをメールで行っていた。今回報告するソフトウェアの研究開発でも、メールを主体にして行っている。

しかし、このように便利なメールであっても、その量が多くなってくると、開発の現状を把握したり、これらの事項がどのような経緯で決定されたかを確認するために、多大な時間と手間を要する。例えば、仕様や設計の変更が頻繁に繰り返されると、どれが最新のものであるのわからなくなったり、仕様や設計のための議論がどうどうめぐりに陥ったりすることがある。これらは、ソフトウェアの効率的な開発を阻害する要因となる。

このような問題点に対して、メールでの議論の推移を簡単に把握し、ソフトウェアの開発をスムーズに進展させるための支援ツールとして、我々はメールによる議論の関係を抽出し、その関係をグラフで表現すると共に、関係するメールの要約を作成するソフトウェアの開発を現在行っている。

本稿では、そのソフトウェアの機能のうち、メールの話題スレッドを要約する方法について、メールを要約する方法の基本方針と、それに基づいて作成したプロトタイプソフトウェアを使った実験結果を示し、実験結果に対する考察を行う。

*The summarization of mails using quotations

[†]Hiroshi Ichiji, Lambda Mathematics and Education Laboratory.

[‡]Jun Kurabe, interervision-razorfish, Inc.

本稿は、以下のような構成になっている。第2節では、テキスト自動要約技術の現状について解説し、その後にはメール要約に関する我々の着目点について述べる。第3節では、我々の要約方法の基本方針と要約実験のあらましについて語る。第4節では、要約実験の結果を示し、その結果に対する考察を行う。第5節では、開発したメール要約ツールについて説明する。第6節では、今後の課題について述べ、第7節でまとめをする。

2 メール要約にむけた検討

2.1 テキスト自動要約技術の現状

本節では、未踏13キックオフ・セミナー¹における奥村氏(東工大)の講演[3]に基づいて、テキスト自動要約技術の現状について簡単に述べる。

要約とは、原文の大意を保持したまま、テキストの長さ、複雑さを減らす処理であると言える。理想的な要約とは、テキストの解釈(文の解析とテキストの解析結果の生成)を行い、テキスト解析結果中の重要部分を要約文として生成することであるが、現状では、テキスト中の重要文(段落)の抽出およびその連結による要約文の生成が行われている。この重要文抽出とその連結による要約生成には、

1. 抽出した文中に代名詞などが含まれている場合、その先行詞が要約文中に存在する保証がない。
2. テキスト中のいろいろな箇所から抽出したものを単に集めているため、抽出した複数の文間のつながりが悪い。
3. 要約中に抽出された文の内容に類似のものがいくつも含まれる可能性がある。

という問題がある。また、テキスト要約において考慮すべき点として、

1. 入力の種類: テキストの長さ、ジャンル、分野、単一/複数テキストのどちらか、...
2. 要約の利用目的: ユーザがどういう人か、要約を何に使うのか、...

indicative 原文の適切性を判断するなど、原文を参照する前の段階で用いられる。

informative 原文の代わりとして用いる。

3. 出力の仕方

があげられる。要約の表示は、通常のテキストとして要約を表示する方法から、ハイパーテキストで表示する方向へ変わりつつある。この理由は、要約が首尾一貫性の欠如からテキストとして可読性に欠けることを、ハイパーテキスト表現によって一まとまりのテキストでないことを明示して回避できることと、ユーザの関心に応じて、自由に要約の長さを変えられるように出来る点にある。

メール(話題)スレッドは、複数の送信者により繰り返される対話であり、その種類は雑多となる。このため、メールスレッドの要約は、複数の人間による雑多な複数対話を対象とした要約となり、非常に難しいものである。また、*indicative* な要約なら有用で可能かもしれないが、*informative* な要約は不可能ではないとも言われている。

なお、メールを対象とした要約の研究は少ない。メールの話題によるスレッド分けを行う研究はいくつかあるものの、要約に関しては比較的少ない。

遠山らの研究[8]では、メールから語彙連鎖とその重み情報、Nグラムモデルによる単語の繋がりにより要旨をまとめた短い討論を出力する手法を提案しているが、これはメール中の重要文の抽出は行っているが、メール全体の内容を理解できるようなものではない。Muresanらの研究[2]では、メールに現れる名詞

¹主催: IPA 未踏ソフトウェア創造事業プロジェクトマネージャ 湯浅太一氏(京大), 開催日: 2001年8月9日, 10日, <http://www.naic.co.jp/jp/m-seminar-report.html>

をテスト文書からの学習に基づいて分類し、それと自然言語の規則から傾向を導き出して、メールの傾向を導き出すものであるが、具体的に要約を導き出すまでには至っていない。Murakoshiらの研究 [4] では、メールに現れる言葉のパターンから、メール相互が肯定、反論などの関係を持つことを導き出しているが、メール内容の要約を行っているわけではない。佐藤らの研究 [5][6] では、ニュース間の関連性を多くの項目で評価し、内容に従ってカテゴリー分けして、ダイジェストを作成している。

2.2 我々の着目点

本節では、我々のメール要約に対する考え方、着目している点について述べる。

我々が研究しているメールの話題スレッドの要約は、要約によってメールの話題の流れをほぼ理解出来ることが目標であり、上で述べた informative な要約であるといえる。また、メールの対話性を保存して要約を行うことも目標にしており、対話性の表現としてハイパーテキストによる出力を行う。

我々は、メールにおいて引用文とそれに対する応答が非常に重要な役割を果たしていると考えている。応答の観点から考えると、引用文の最後の1文、あるいは最後の2文くらいが特に重要であり、この部分に対して引用文の後の返答に相当する部分が書かれていることが多いと予想している。このことから、引用文を中心にして要約を行うこととする。

また、引用文は現在着目しているメールより以前に発信されたメールの中に存在しているものであり、引用文を主体にして要約を行うことは、メールの内容の繋がりや対話性を保持する上でも非常に重要であると考えている。引用文を繋げていくことで、メールの話題のトレースが出来ると予想されるが、引用文を全て取り込むと多くの場合には要約文が長くなりすぎるので、引用文をどれだけ省略するかも、要約作成の際には重要となるであろう。

またメールの中で重要な話題には、キーとなる単語が繰り返し登場すると予想されるので、登場回数の多い単語を含む文を重要な文であると見なして、要約の中に取り込むことが必要となる。この場合、どのくらいの数をキーとなる単語とするか、その個数を決めることも要約の質を決める上で重要な要素となる。キーとなる単語が少ないと、本当は重要なのに要約から落ちてしまう文が出現する可能性があるし、キーとなる単語が多すぎると、重要でない文が大量に要約文の中に入り込み、要約文の冗長性が増してしまう。

2.3 我々の要約の方針

以上の観察に鑑み、さらにいくつか手作業で実験を行い、我々は次のような要約生成の方針を立てた。

1. 話題が関連するメールは、発信順が遅いメールから早いメールへ順番に要約を行う。
2. 引用文は引用ブロックの最後の1文を要約文に採用する。
3. 引用ブロックの後は、直後の3文を要約文に採用する。ただし、3文中にその後の引用ブロックが出現したときには、引用ブロックの前の文までを要約文に採用する。
4. 重要文抽出のためのキーとなる単語には、名詞（ただし代名詞は除く）で出現数が上位3位までの単語を採用して、それを含む文を要約文に採用する。
5. すでに要約文中に現れた引用文が、要約対象メール中に現れているときには、その文を要約文に採用する。

3 メール 요약

3.1 요약 소프트웨어의作成

上の 요약方針に基づいて、我々は 요약 소프트웨어を作成した。 요약 소프트웨어は、そのユーザーインターフェイスも含めて全て Java 言語でアプリケーションとして作成し、形態素解析には茶筌 [1] を利用している。

3.2 요약に使った題材

今回は、 요약の評価を行うための題材として、本稿で記述しているソフトウェアを開発している際に交わしたメールと未踏ソフトウェア湯浅グループの ML に流れたメールを使った。ソフトウェア開発を支援するツールと銘打っている我々のプロジェクトにとっても格好の題材である。

3.3 요약例

2つの対象メールの本文と 요약 소프트웨어による 요약文を示す。以下の「対象メール (1)」に対する 요약文が「対象メール (1)の 요약」であり、「対象メール (2)」に対する 요약文が「対象メール (2)の 요약」である。

3.3.1 対象メール (1)

```
> From: Jun-Krb <jun-krb@mars.dti.ne.jp>
> Subject: Re: [mito] Windows2000 で動かすことができました .
> Date: Fri, 04 Jan 2002 17:16:33 +0900 (JST)
>
>> From: Hiroshi Ichiji <ichiji@acm.org>
>> Subject: Re: [mito] Windows2000 で動かすことができました .
>> Date: Fri, 04 Jan 2002 15:31:22 +0900 (JST)
>>
>>> From: Jun-Krb <jun-krb@mars.dti.ne.jp>
>>> Subject: [mito] Windows2000 で動かすことができました .
>>> Date: Fri, 04 Jan 2002 15:27:19 +0900 (JST)
>>>
>>>> まだ、98 上で可能なのは、確認していません .
>>>>
>>>> あとで ftp できるところに full set を置いてください .
>>>> あと環境設定のガイドも .
>>>> Windows 98 と Me のマシンで確かめてみます .
>>>>
>>> 自宅の 98SE マシンで確かめましたが、動きません .
>>> chasen が環境設定ファイル chasenrc を読むところで、なぜかファイルパ
>>> スを認識できずに、no such file or directory エラーとなります .
>>> 何度も確かめていますので、include と import のミスのようなことはありません .
```

- > > 不思議だ...
- >
- > ファイルパスはどこで指定しているのでしょうか？
- > プログラムの中，それとも環境変数などを使っている？
- > Windows NT系 (Windows NT/2000) と Windows 95系 (Windows 95/98/Me) では，環
- > 境設定の仕方が微妙なところで違ったりします．
- > そこあたりを確認する必要があります．

ファイルパスは，環境変数で設定しています．

Windows98 上でも，与えたファイルパスを入手し，ファイルを読みに行っているのですが，no such file or directory エラーとなっています．設定したファイルパス名は，その通りエラーメッセージとともに表示されているので，渡すまではうまく行っていると思われます．

Windows2000 と同じ設定で，だめでしたので，相対パス，絶対パス，パス区切りを"/"，"\"，"\"にするなどしてみましたが，だめでした．

//倉部

3.3.2 対象メール (1) の要約

- > > > あと環境設定のガイドも．
- > > chasen が環境設定ファイル chasenrc を読むところで，なぜかファイルパ
- > > スを認識できずに，no such file or directory エラーとなります．
- > ファイルパスはどこで指定しているのでしょうか？
- > Windows NT系 (Windows NT/2000) と Windows 95系 (Windows 95/98/Me) では，環
- > 境設定の仕方が微妙なところで違ったりします．
- > そこあたりを確認する必要があります．

ファイルパスは，環境変数で設定しています．

Windows98 上でも，与えたファイルパスを入手し，ファイルを読みに行っているのですが，no such file or directory エラーとなっています．設定したファイルパス名は，その通りエラーメッセージとともに表示されているので，渡すまではうまく行っていると思われます．

Windows2000 と同じ設定で，だめでしたので，相対パス，絶対パス，パス区切りを"/"，"\"，"\"にするなどしてみましたが，だめでした．

3.3.3 対象メール (2)

- > xxxxxxxx xxxxxxxx さんは書きました：
- > >XXX です．
- > >
- > >> E -> E * F だと

```

> >
> >E -> E * F | F
> >ですね?
>
> はい。
>
> >> ----
> >> command_stack:
> >> command_stack: 10,
> >> syntax trace : value 10
> >> command_stack: _F_,
> >
> >この次に
> >command_stack: _E_,
> >ならないのは何故でしょう?
>
>         // _E_ -> _F_
>         if ( stack_tbl.m_command_stack.size() > 0
>             && stack_tbl.m_command_stack[0] == _F_
>             && next != MUL
>             && next != DIV
>             )
>         {
>             stack_tbl.m_command_stack[0] = _E_ ;
>             continue ;
>         }
>
>     ここで、_F_ の後ろを先読みして next == MUL を検出しているのに、_E_ にはならないのですが。
>     あれ?、この条件が間違っているのかな?^^;

```

ここが問題のようですねえ .

3.3.4 対象メール (2) の要約

```

> >XXX です .
> >この次に
> >command_stack: _E_,
> >ならないのは何故でしょう?
>
>     ここで、_F_ の後ろを先読みして next == MUL を検出しているのに、_E_ にはならないのですが。
>     、この条件が間違っているのかな?^^;

```

ここが問題のようですねえ .

表 1: メールの要約率

番号	本文 (文の数)	要約文 (文の数)	要約率 (要約文/本文) (%)
1	68	38	55.8
2	42	32	76.1
3	23	18	78.2
4	30	20	66.6
5	74	46	62.2
6	53	39	73.5
7	81	50	61.7
8	327	27	8.2
9	59	38	64.4
10	32	9	28.1

3.4 要約率

要約率を、要約対象メールの本文の文数 (但し空行は数えない) に対する要約文の文数と定義すると、今回実験したメール文書では表 1 のようになる。また、2 番のメールでは意味的に見たときに重要だと思われる一文が要約文から欠落している。

4 実験に対する考察

本節では、実験結果を我々の目標の観点から考察する。考察を行うポイントは、

- informative な要約になっているか、
- プレーンテキストレベルでの対話性を保存しているか、
- 登場回数の多いキーとなる単語が重要文を抽出するために有効に機能しているか、

である。

4.1 要約の質

まず、実験で行った要約が informative な要約になっているかを考察する。第 3.3 節の 2 例に代表されるように、実験を行った 10 通のメールに関する限り、要約文を読んだだけで内容を十分に理解できる。その意味で要約は informative と言えるが、表 1 からわかるように要約率が 60% 台のものが多く、要約の圧縮が低いことから、要約文で内容を理解できるのは当然の結果と言えないこともない。

4.2 対話性

要約文に、引用文を引用ブロックの最後の 1 文、引用文の直後の 3 文を採用するルールにより、要約文において対話性が十分に保たれていることが確認できた。後で述べる重要文の抽出により、引用ブロックの最後の 1 文でない引用文も要約文中に現れ、メールにおける引用文を用いた対話性に関して十分な結果を得ていると言える。

試しに要約生成のルールを、引用文を引用ブロックの最後の 1 文、引用文の直後の 2 文を採用するに変えて要約実験すると、要約における対話性を保存するために必要な文の欠落が多々見られる。引用文の直後の 3 文を採用するというルールは、要約の対話性に関して微妙なバランスを保つものであると考えることが出来る。

引用ブロックから要約に採用する文を終わりの 2 文、3 文とルールを変更して実験すると、今度は要約文が長くなりすぎ、要約文と原文の差がほとんどなくなってしまふ。引用中の重要な文は重要文抽出で現れてくるので、引用文の要約への採用は引用ブロックの最後の 1 文を選ぶことが最適であるように今回の実験からは感じた。

4.3 重要文抽出

出現数が上位 3 位までの名詞を使つての重要文抽出は、第 3.3 節の対象メール (1) の要約文でよくわかるように、ファイルパスに関する議論が再現されている。対象メール (1) の要約の場合には、キーとなる単語として抽出されたのが「ファイル」、「パス」、「環境」である。これらの単語を 1 つでも含むものは重要文と判定して、要約に取り入れるようにしているので、どうしても重要文の数が多くなり、要約文が長くなってしまふ原因にもなっている。

キーとなる単語で重要文を抽出する場合には、キーとなる単語をいかに減らすかが重要な課題となる。しかし、出現数が 2 位までの単語をキーとするように変更して実験すると、要約文は急に短くなるが、意味的に重要と思われる文が要約から欠落し、informative な要約とならなくなってしまうという問題が新たに発生する。

5 メール要約ツール Preface

本節では、これまでに述べたメールの要約機能を入れたメール要約ツール Preface について簡単に説明する。

5.1 メールに関連性抽出

メールの話題スレッドを要約するためには、メールの関連性を抽出することがまず必要になる。

メールの関連性抽出とは、どのメールが他のメールの返事になっているかの関係を抽出するものであり、有効グラフとしてメール間に関係を付ける。

メールの関連性は、主にメールヘッダーの Message-id, In-Reply-To, References 情報から得る。これらの情報が完全な形で存在すれば関係付けはかなり容易であるが、メイラーによっては In-Reply-To の情報がなかったりする。また、メールの返信を書くときに、返事を出すメールに対して返信モードで書かず、わざわざ新規のメールを作って返信のメールを出す人もいる。この場合には、メールヘッダーの情報からではメールの関係を抽出できない。メールヘッダーから関連性を抽出できないときには、本文中の引用文を情報として、どのメールへの返信であるかを推測する。

メールの関連性を表す木を生成するアルゴリズムは以下の通りである。

1. メール A の In-Reply-To の ID がメール B の Message-ID と一致するときに、メール B をメール A の親とする。
2. メール A が In-Reply-To を持たず References を持っているとき、References のリストにあるメールのうち最新の発信日時を持つメール B をメール A の親とする。

3. メール A が In-Reply-To と References を持たず本文中に引用文が存在するときに、メール A と”Re:”やメイリングリストの番号表示を除いて同じ Subject を持つメール B において、メール A の引用文がメール B で引用文としてでなく現れるときに、メール B をメール A の親とする。
4. 以上によって求められるものだけが、メールの親子関係である。

5.2 ユーザインタフェース

メール要約ツール Preface のユーザインタフェースは以下のようになっている。

図 1 はメイン画面である。左側のサブウィンドウにメールフォルダーのリスト表示、右側の一番上のサブウィンドウに選択されたメールフォルダー内のメールの発信日時、発信者、タイトルが表示されている。選ばれたメールの内容がその下のサブウィンドウで表示され、一番下のサブウィンドウが要約の表示のための領域である。上部にあるボタンは、左からメールの関連図を生成する「Graph ボタン」、要約を生成する「Summary ボタン」、メール関連図上で選択されたメールから要約を生成する「Summary from Graph ボタン」である。

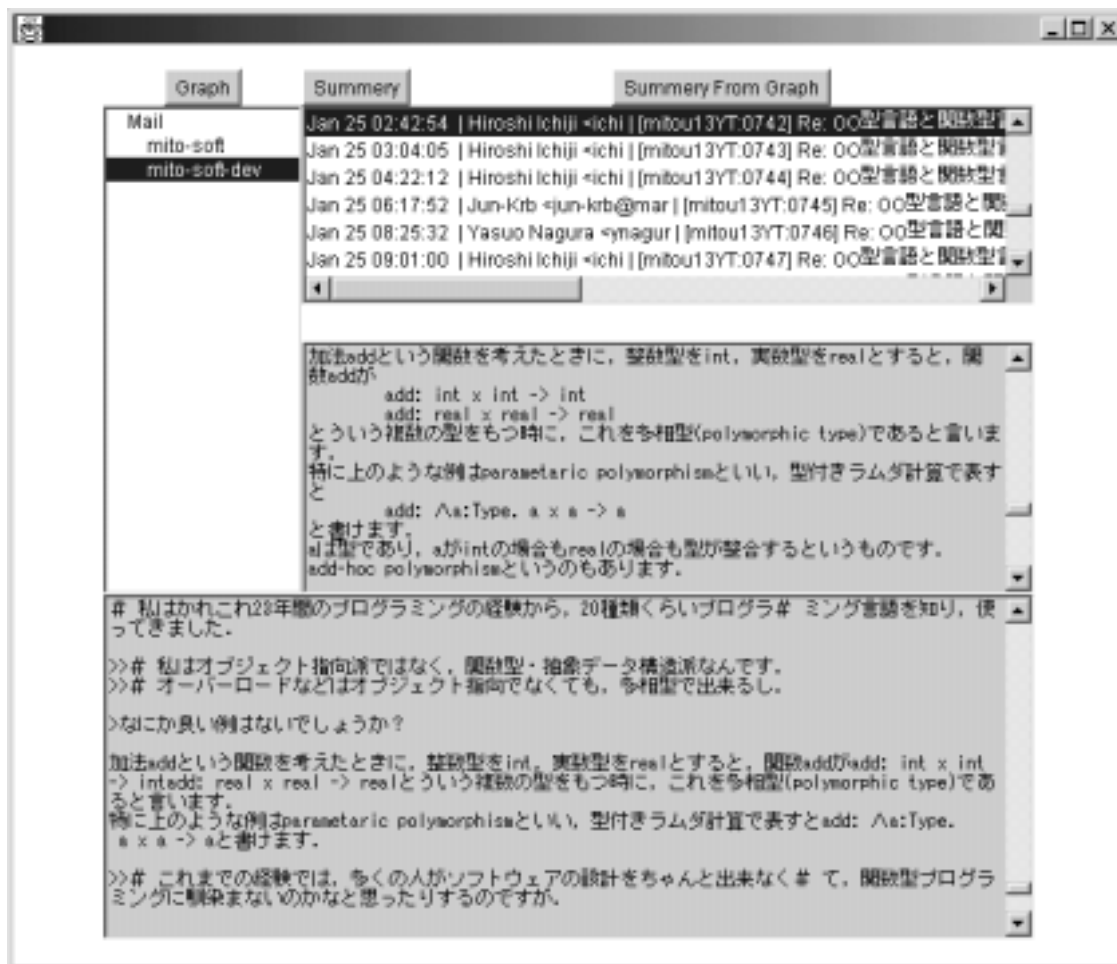


図 1: Preface のメール要約画面

メールを選択して図1の「Graph ボタン」を押すと、メール関連性の計算が上記のアルゴリズムで行われ、図2のようなメール関連図を生成する。メールを選択して図1の「Summary ボタン」を押すと、メールの関連性を計算した後に、メールの要約を行いサブウィンドウに結果を表示する。



図 2: Preface のメール関連図

図2からわかるように、メールの関連性はグラフで表現しており、左上の表示領域を直接操作するか、スクロールバーを使って表示する領域を変えることができる。それぞれのメールはノードとして表され、タイトルと発信者が表示されており、発信者ごとにノードの色を変えている。ノードを選択して、図1の「Summary from Graph ボタン」を押すと、メールの要約を行い結果が表示される。また要約の結果はHTML ファイルとしても生成し、引用箇所へジャンプしたり、メールの原文との行き来をすることも可能になっている。

6 今後の課題

今後の課題は

- 要約アルゴリズムの評価
- informative な要約,
- 登場回数の多いキーとなる単語による重要文の抽出,
- キーとなる単語の選択

のそれぞれにある。

現在の要約アルゴリズムは、多種多様なメールに対して検討を加えた上で予想して導き出したものであるが、実験対象としたメールの特徴から偶然にも比較的上手くいっている可能性がないとはいえない。このアルゴリズムが有効であることを客観的に示す方法を見出すことが課題となる。

今回の実験では要約文の長さが一般に長めとなっている。しかし、一般の人が要約に抱くイメージはもっと短く簡潔なものである。現在の我々の方法では、要約を短くすると informative 性を損なう結果となる。短くて informative な要約を作るためには、重要文を接続するだけでなく、自然言語処理を使って数文の余計な箇所を 1 文にまとめなおすような工夫が必要となる。

重要文を抽出するためのキーとなる単語の個数を、今回の実験では一定にしているが、この方法では短いメールにおいて登場回数が 3 位までの単語が多数登場し、メール中のほとんどが重要文になってしまう。要約対象のメールの長さに応じて、重要文を抽出するためのキーとなる単語の数を変えることが課題である。

キーとなる単語は形態素解析を行う辞書に依存してしまい、辞書に載っていない重要な単語が、今回の実験ソフトウェアでは重要単語として選択されないという問題もある。この問題に対しては、豊橋技術科学大学梅村研究室で開発が行われているキーワード抽出ソフトウェア [7] の技術を用いて辞書に単語登録などを行い、辞書に依存しない重要単語の選択方法を確立することが課題である。

7 まとめ

メールの要約ソフトウェアを開発し要約実験を行った。要約作成に、引用文は引用ブロックの最後の 1 文を採用、引用ブロック直後の 3 文を採用、名詞で出現数が上位 3 位までの単語をキー単語としてそれを含む文をに採用するという方法を取ることで、informative で対話性のある要約を生成することが出来ることがわかった。同時に、要約文の長さ、重要文抽出のための単語の選択方法に問題があることが明らかになった。

謝辞

メール要約実験のために、未踏ソフトウェア創造事業湯浅太一 PM グループのメイリングリストに流れたメールを使わせていただくことを許可していただきました皆様に感謝します。本研究は、情報処理振興事業協会 (IPA) の平成 13 年度未踏ソフトウェア創造事業の支援を受けています。

参考文献

- [1] <http://chasen.aist-nara.ac.jp/index.html>.ja
- [2] Smaranda Muresan, Evelyne Tzoukermann, and Judith L. Klavans: Combining Linguistic and Machine Learning Techniques for Email Summarization, In Proceedings of CoNLL 2001 Workshop at ACL/EACL 2001 Conference, Toulouse, France, July 2001.
- [3] 奥村 学: テキスト自動要約技術の動向, 未踏 13 キックオフ・セミナー (2001 年 8 月 9 日, 10 日) 資料.
- [4] Hiroyuki Murakoshi, Akira Shimazu, Koichiro Ochimizu, Construction of Deliberation Structure in E-mail Communication, International Journal of Computational Intelligence, 16, 4, pp.570-577 (2000).
- [5] 佐藤円, 佐藤理史, 篠田陽一: 電子ニュースのダイジェスト自動生成, 情報処理学会論文誌, Vol.36, No.10, pp2371-2379 (1995).

- [6] 佐藤理史, 佐藤円: ネットニュースグループ fj.wanted のダイジェスト自動生成, 自然言語処理, Vol.3, No.2, pp19–32 (1996).
- [7] 田中路子, 武田善行, 仲村大也, 山本英子, 梅村恭司: 統計処理によるキーワードの抽出実験, 第 42 回プログラミング・シンポジウム, pp.155–158 (2001).
- [8] 遠山義洋, 西田豊明: 話題構造の抽出と変形による対話録の自動要約, 第 14 回人工知能学会全国大会 (2000) .