文書構造の認識と言語の特徴の利用に基づく電子メールからのスケジュールとToDoの抽出

長谷川 隆明† 高木 伸一郎†

インターネットの普及にともない，電子メールはコミュニケーションの主要な手段となった．一方，Personal Information Manager（PIM）ソフトウェアの普及にともない，個人情報を計算機で管理するユーザが増えている．ユーザの個人情報として，アポイントメントの日時や場所等のスケジュールや，期限をともなう電子メールの返信等のToDoがあげられる．しかしながら，電子メールにより伝達されるスケジュールやToDoに関する情報の管理は，これらの情報を含む文書の整理やPIMソフトウェアとの連携の際に，電子メールを受信するユーザの人手を必要としていた．本稿では，ユーザが受信した電子メール文書からユーザに伝達されるスケジュールとToDoの情報を抽出する手法を提案する．電子メール文書は，任意の目的への使用と自由な形式による情報伝達のため，文書構造や言語表現が一様ではない．本手法の特徴は，スケジュールやToDoを含む電子メール文書の構造と言語の特徴に着目したレイアウト情報とパターンマッチングを用いた，文書構造の認識と情報抽出および情報の関連付けである．電子メールの実文書を対象とした抽出実験により，電子メールのフィルタリングやPIMソフトウェアへの入力等の実用に耐えうる高い精度で，スケジュールとToDoを抽出できることを示す．

Extraction of Schedules and To-Do Items from E-mail Messages by Identifying Message Structures and Using Language Expressions

TAKAAKI HASEGAWA† and SHIN'ICHIRO TAKAGI†

As the Internet has become popular, e-mail is now an important means of communication. On the other hand, as the Personal Information Managers (PIM) applications have come into wide use recently, many users manage their schedules, such as event date and event location, and to-do items, such as answers to e-mail messages from someone by the appointed time, with their computers. However, a problem is that e-mail receivers cannot easily sort out messages with these information from many incoming messages and build up a close connection with the receivers' PIMs. Therefore, our goal is extracting these information from the e-mail messages users receive. E-mail is open to any purpose and any format. So these information is not formalized, and message structure and language expression are not uniform. We reveal the characteristics of the structure and language used in e-mail messages and propose a way to identify the structure and extract information by using layout information and pattern matching and relate matched partial information with components of these information. Extraction evaluations demonstrate high recall and precision. Our proposal can be put to practical use, such as filtering messages and inputting the information to PIMs.

1. はじめに

インターネットの爆発的な普及により，多くのユーザが電子メールを用いてコミュニケーションを行うようになった．電子メールは，多数の人に同時に情報を伝達する同報性を有するため，送信するユーザにとって有効なコミュニケーションの手段である．また，携帯電話等の無線通信のインフラが整ったため，ユーザは外出先からでも電子メールを送受信することにより，時間と場所を選ばない非同期コミュニケーションが可能となった．一方，Personal Information Manager（PIM）ソフトウェアの普及により，アポイントメントやイベントのスケジュールや指定された期限までの電子メールの返信等のToDoといった個人情報を計算機で管理するユーザが増えている．電子メールによるコミュニケーションの中でも，電子メールにより伝達されるスケジュールや期限をともなう返信の依頼

† NTTサイバースペース研究所
NTT Cyber Space Laboratories

等の情報を計算機で管理することは，ユーザあるいはユーザを代行するソフトウェアにとって重要なタスクとなっている．

しかしながら，電子メールを受信するユーザは，1人あたりの受信する電子メールの文書数が増大した結果，いわゆる情報洪水の中に身を置くことになった．このような状況では，早急に返信を要する文書を受信していたとしても，多量に受信した文書の中から真っ先に読むべき重要な文書を見つけることは困難である．また，ユーザに関係のある実世界のイベントやアポイントメントの情報を含む文書をユーザが受信しても，その情報を定型化して PIM ソフトウェアへ入力するにはユーザの人手に頼らざるをえない．このように，電子メールによるコミュニケーションは，送信側にとっては同報性と非同期性の点から都合が良いが，受信側にとっては情報の整理や PIM ソフトウェアとの連携の点において課題を残している．

本稿では，電子メールで伝達されるアポイントメントやイベントの日時や場所等の情報をスケジュール情報，返信の依頼やそれに関する期限や連絡先の情報を ToDo 情報と定義する．我々は，電子メールによるコミュニケーションにおいて，これらの情報は普遍的に重要な情報であると考える．なぜならば，電子メールを受信するユーザの職種，関心事，対人関係等によって重要な情報は様々であるが，電子メールが主にコミュニケーションの手段として用いられる以上，アポイントメントの通知や受信するユーザに対する依頼はすべてのユーザに対して重要な情報の1つであると考えられるからである．しかしながら，スケジュール情報や ToDo 情報は，必ずしもユーザが受信する電子メールの文書に存在するとは限らず，位置や順序，個数，表現形態も多様である．このため，電子メールによって伝達されるスケジュール情報や ToDo 情報の抽出は，単純なパターンマッチングでは不十分であり，また具体的な手法の提案は行われていなかった．

本稿では，電子メールの整理や電子メールと PIM ソフトウェアとの連携を実現するために，ユーザが受信する電子メールからスケジュール情報と ToDo 情報を抽出することを目標とする．電子メールは，任意の目的や任意の形式を許すため，スケジュール情報や ToDo 情報は様々な文書構造や言語表現により伝達される．本稿では，これらの情報を伝達する電子メール文書の構造と言語の特徴に着目し，レイアウト情報とパターンマッチングを用いた文書構造の認識と情報抽出，および，文書構造の種別と抽出した情報の存在位置を用いた情報の関連付けについての手法を提案する．

本稿の構成は以下のとおりである．2章で従来の情報抽出の研究について述べる．3章で電子メール文書を対象としたスケジュール情報と ToDo 情報を抽出するアルゴリズムを提案する．4章で電子メールの実文書を対象とした抽出実験の結果を報告する．5章で本手法の有効性と課題を考察し，6章で結論を述べる．

2. 従来研究とその課題

ネットワーク上に存在する主なテキスト文書のソースは，電子メール，電子ニュース，World Wide Web（WWW）があげられる．これらはユーザがだれでも情報を発信できるという特徴を持つ．しかし，発信する相手を基準にとれば，電子メールは不特定多数を対象とする電子ニュースや WWW と異なり，親しい間柄の仲間同士や特定の知人同士のコミュニケーションに使われるという特徴を持つ．このため，電子メール文書には，略語や会話調の表現などの非一般的な表現や，文法の誤り，漢字変換の誤り，誤字脱字等のノイズが含まれることが多く，電子メール文書のテキストの質は新聞記事等に比べて悪い．

スケジュール情報や ToDo 情報の抽出は，テキストから中心となる情報を取り出す情報抽出の一種である．情報抽出の研究では以前から主に新聞記事を対象とした情報抽出のコンテスト MUC（Message Understanding Conference）[8] が開かれ注目を集めてきた．情報抽出や重要文抽出の手法として，構文解析と意味解析を用いた手法[6),10)] や，形態素解析とパターンマッチングを用いた手法[3),5)] が報告されている．有限状態オートマトンを用いた FASTUS[1),2)] は，ドメインに依存する特定の単語から句を同定し，その結果を入力として辞書的情報を使ったパターンとのマッチングを行う．これらはいずれも辞書を用いた解析を前提としている．しかしながら，テキストの質が悪い電子メール文書に対して辞書を用いた解析を行うと，解析結果が未知語として処理されたり解析誤りを起こしたりする．そのため，解析結果を有効に利用できない．

形態素解析を行わずパターンマッチングのみを用いる手法としては，新製品紹介の新聞記事からパターン表現を用いて未知語である製品名や販売元等を抽出する手法[7)] や，特許広報から特定のパターンを含む範囲を抄録として抜き出す手法[4)]，時制や文のタイプ，修辞構造等を表す表層表現を手掛かりにして記事中の文の重要度を決定する手法[12)] が報告されている．これらの手法は，主に新聞記事等の校正された文書を対象としているので，意図的に定型化された文書や，目的や対象が限定された文書に対しては有効である．しか

しながら，電子メール文書は任意の目的と任意の形式で任意の内容を含むので，これらの手法をそのまま適用するのは難しい．さらに，これらの手法はレイアウトを考慮していないので，レイアウトを用いて構造化された文書に対しては，レイアウトの情報を利用できない．

レイアウトを考慮する手法としては，会議の告知や論文募集に関する電子ニュース記事からダイジェストを作成する手法[9]が報告されている．この手法では，センタリング等の行スタイル情報と特定のキーワードやパターンによるパターンマッチングとを用いることによって，サマリとなる情報を抽出している．しかしながら，対象となる文書は目的や対象が限定され，抽出すべき情報が必ず1つ存在することが前提となっている．また，レイアウトが使われず複数行にわたって記述されている文章中の情報や，開催日時や場所といった必須の情報だけでなくあらかじめ特定できない情報を抽出することはできない．さらに，箇条書きのラベルがスタイルを考慮して分かち書きされている場合には，行スタイル情報は正しい箇条書きラベルを特定できず，箇条書きから情報を抽出できない．

そこで，本稿では，箇条書きの特定による文書構造の認識と言語の特徴に基づくパターンマッチングを用いた情報抽出の手法を提案する．本手法の特徴は，文字列の長さや位置というレイアウト情報と，辞書的情報やドメイン依存のキーワードでなく発話行為[13]に着目した言語の特徴とを用いることである．

3. 電子メール文書を対象とした情報抽出

3.1 文書構造と言語表現

スケジュール情報やToDo情報を含む文書のスタイルやドメインは，**図1**に示す例からも明らかなように一様ではない．しかしながら，電子メール文書からスケジュール情報やToDo情報を抽出するには，電子メール文書にスケジュール情報やToDo情報がどのような形式でどんな表現で記述されているのかという文書の構造と言語の特徴をつかむ必要がある．そこで我々は，実際に我々が受信した電子メール文書の中から日時表現を含む80通を選んだ．これらの内容は，打合せのアポイントメントや講演会，送別会，結婚式等の各種イベントの案内とイベントへの出席の可否，質問に対する回答の返信依頼である．これらの中から日時と場所が明記されているスケジュール情報を含む65通と，期限が記述されている29通とそれを含めて返信の依頼が明記されているToDo情報を含む51通を調べた．その結果，物理的な文書の構造に関しては，

```
    0   4   8  12  16  20  24  28  32  36  40  44  48  52  56
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---
  | Subject: GOLF
  | From: yamada@abc.co.jp
  | To: golfers@abc.co.jp
  | Date: Wed, 13 May 1998 20:00:49 +0900
  |
 1| 第3回小松部長杯ゴルフコンペを下記のとおり開催いたします．
 2| みなさま奮ってご参加いただきますようよろしくお願いいたします．
 3| なお，詳細は別途連絡いたします．
 4|
 5| 1．日　　時　　平成10年6月8日（月）
 6|                  エメラルドコース　8：00スタート
 7| 2．場　　所　　葉山国際カンツリー倶楽部
 8|                  [三浦群葉山町上山口　Tel0468-12-1234]
 9| 3．参 加 費　　約25，000円（昼食別）
10| 4．ハンディ　　新ペリア方式にて算出
11|
12| 幹事：ご質問等は山田 (yamada@abc.co.jp) まで
13|
14| 山田 太郎 (yamada@abc.co.jp)
```

(a) 電子メール文書の例1

```
    0   4   8  12  16  20  24  28  32  36  40  44  48  52  56
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---
  | Subject: 講習会のお知らせ
  | From: Hisashi OOHARA < oohara@abc.co.jp >
  | To: all@abc.co.jp
  | Date: Wed, 12 Nov 1997 13:53:44 +0900
  |
 1| 大原です．
 2|
 3| 12月10日（水）15：00～16：30に104C会議室で
 4| 前回未受講の社員を対象とした電子決裁システム講習会が行われます．
 5| 未受講の人は，やむを得ない事情がない限り出席せよとのことです．
 6|
 7| 各位受講実績と参加の可否（否の場合は理由を添えて）を大原まで
 8| 11／18日中にご回答ください．
 9| よろしくお願いします．
```

(b) 電子メール文書の例2

図1 スケジュール情報とToDo情報を含む電子メールの例
Fig. 1 E-mail messages with schedules and to-do items.

表1 スケジュール情報とToDo情報を記述する表現形態の内訳
Table 1 Statistics of typical expressions for schedules and to-do items.

分類	日時	期限
箇条書き	50 (76.9%)	2 (6.9%)
文章	15 (23.1%)	27 (93.1%)

記述された日時や期限の表記の仕方が“日時：”などによるラベル付きの箇条書きとそうでない文章の部分に分けることができた．分類した結果を**表1**に示す．ラベル付きの箇条書きを用いず文章として記述している表現は，スケジュール情報よりもToDo情報の方に多く見られた．少なくとも日時表現はドメインには依存しないので，この傾向には一般性があると考えられる．

また，**表2**にスケジュール情報とToDo情報を記述している表現に見られる言語の特徴を示す．表中の項目にある見出しの例として，“〈イベント〉について”，“〈イベント〉のご案内”等が見られ，発話行為なしの例としては，“○月×日の〈イベント〉ですが”，“ご案内しました〈イベント〉の件ですが”等の表現が見

表 2 スケジュール情報と ToDo 情報を記述する言語の特徴
Table 2 Characteristic of languages expressing schedules and to-do items.

	表現	頻度	割合
スケジュール情報	“開催します”	26	40.0%
	“行います”	7	10.8%
	“お知らせします”	5	7.7%
	“決定しました”	3	4.6%
	“あります”	3	4.6%
	その他の発話行為	7	10.8%
	発話行為なし	9	13.8%
	見出し	4	6.2%
	その他	1	1.5%
(合計)		65	100 %
ToDo 情報	“連絡下さい”	27	52.9%
	“お知らせ下さい”	5	9.8%
	“送って下さい”	4	7.8%
	“報告下さい”	4	7.8%
	“回答下さい”	2	3.9%
	その他の発話行為	9	17.6%
(合計)		51	100 %

```
<!ELEMENT body 0 0 (own | ref)*>
<!ELEMENT own - - (item | sentence)+>
<!ATTLIST own p_line NUMBERS #REQUIRED
              s_point NUMBER #REQUIRED>
<!ELEMENT ref - - (item | sentence)+>
<!ATTLIST ref p_line NUMBERS #REQUIRED
              s_point NUMBER #REQUIRED>
<!ELEMENT item - - (label)>
<!ATTLIST item l_line NUMBER #REQUIRED>
<!ELEMENT label - - (#PCDATA)>
<!ATTLIST label type (date |
                      location |
                      deadline |
                      contact |
                      others) #REQUIRED
                name CDATA #REQUIRED
                l_point NUMBER #REQUIRED
                length NUMBER #REQUIRED>
<!ELEMENT sentence (#PCDATA)>
<!ATTLIST sentence l_line NUMBER #REQUIRED>
```

図 2 文書に付与するタグの定義
Fig. 2 Definitions of tags attached to messages.

られた.

3.2 抽出処理の過程

文書構造の認識と言語の特徴を用いる本手法は，大きく分けて以下の 3 つの過程からなる.

(1) 箇条書きの特徴を用いた文書構造の認識
(2) ラベルの属性を用いた箇条書きからの情報抽出と言語の特徴を用いたパターンマッチングによる文章からの情報抽出
(3) 抽出した情報が持つ文書構造の種別と情報の存在位置を用いた情報の関連付け

構造認識の過程において，SGML を用いてタグを挿入する．タグの定義を**図 2** に示す．構造認識と情報抽出の過程では，電子メール文書の構造と言語の特徴を用いて作成したパターンによるパターンマッチングを用いる．**図 3** にパターンの例を BNF 記法風に示す．記号 “{a|b}” は “a または b” という意味を表す．“[ ]” は文字の種別の集合を表す．“( )?” は省略可能を，“+” は 1 個以上の繰返しを，“*” は 0 個以上の繰返しをそれぞれ表す．“$” に続く記号は変数名で，任意の文字列とマッチし，マッチした文字列を格納する．個々の詳細のパターンがどのような働きをするかは以下で論じる.

3.3 文書構造の認識

3.3.1 本文と引用文の分離

電子メールにおいては，他人のメッセージを引用するフォワードやリプライが用いられる場合がある．そのような場合，スケジュール情報が引用されたメッセージにあることも多い．そこで，引用文を除いた本文から情報が抽出されない場合には，引用文からの情報抽出を行うことが必要になる．また，リプライの場合は引用文が本文と交錯して挿入されることが多い．そのため，電子メール文書の構造を認識するうえで，本文と引用文の範囲を判別することが不可欠である.

引用文には，“⟩” や “⟩⟩” 等の引用を表す特定の記号が行頭に付けられていることが多い．これらの記号から構成されるパターンを用いることにより，引用文の範囲を特定することが可能である．ここで，電子メールでは文章の途中でも改行が挿入されることが多いので，行の概念を明確にするために物理行と論理行という 2 つの行を定義する．物理行は図 1 に示すようにメッセージが表示された状態における位置とし，論理行は文章の途中にある改行の削除により接続される文のような論理的な単位の順序とする．すべての物理行に対して，本文か引用文かの種別を示すタグを付けることによって，電子メール文書を本文と引用文とに区分する．電子メール文書の n 行目が本文であれば，タグ ⟨own p_line=‘n’ s_point=‘*#point*’⟩ を付ける．属性 p_line は物理行の番号を表す．属性 s_point は文字列の開始位置を表し，行頭から空白の連続する長さとする．空白がなければ，属性 s_point は 0 と定義する．また，タブは 8 byte の空白とする．#が先頭に付加された文字列は，文書から得られる文書固有の数値や文字列を表す．n 行目が引用文であれば，タグ ⟨ref p_line=‘n’ s_point=‘*#point*’⟩ を付ける．文字列の開始位置は，引用記号の文字列の長さと引用記号から空白の連続する長さの和とする．文字列の開始位置より後の文字列が，これらのタグで囲まれる.

```
<Itemization>   ::=  {(<Prefix>)?<Label1>(<Suffix>)?<Blank>*(<$Content>)?
                     |<Prefix><Label2>(<Suffix>)?<Blank>*(<$Content>)?
                     |<Label2>(<Suffix>)?<Blank>*<$Content>}
<Prefix>        ::=  {"○"|"◎"|"●"|"・"|[1-9]+({"."|","})?}
<Suffix>        ::=  ":"
<Label1>        ::=  < 空白以外の文字 >{< 任意の文字 >*< 空白と数字以外の文字 >}{1,n}
<Label2>        ::=  < 空白以外の文字 >*< 空白と数字以外の文字 >
<$Content>      ::=  < 任意の文字 >+
<Blank>         ::=  {< 空白 >|< タブ >}
```

(a) ラベル付き箇条書きのパターン

```
<Schedule>      ::=  {(<Start_rep>(<End_rep>)?)?(<Location_rep>)?
                     <Event_rep><Object_pp><Inform>
                     |(<Start_rep>(<$Location+>)?"の")?
                     <$Event+>("の件")?{"について"|"です"}}
<Inform>        ::=  {"開催"{({"致"|" いた"})?{"します"|"したい"}|" する"}| "行"("な")?"われ"{"ます"|"る"}}
<Event_rep>     ::=  (<Modify_rep>)?<$Event+>
<Modify_rep>    ::=  < 任意の文字 >+{"とした"|"についての"}
<ToDo>          ::=  {(<Deadline>)?(<Contact>)?<Request>|<Deadline>}
<Request>       ::=  ({"ご"|"御"})?{"回答"|"連絡"|" 報告"}("を")?
                     {{("して")?{"下さ"|"くださ"|"頂"|"いただ"}}|{"願い"|"ねがい"}}
<Start_rep>     ::=  <Date_pat>({"から"|"～"|"より"|"スタート"|"-"})?
<End_rep>       ::=  <Date_pat>({"まで"|"に"|"にかけて"})?
<Location_rep>  ::=  <$Location+>{"におきまして"|"において"|"にて"|"で"}
<Object_pp>     ::=  {"を"|"について"|"が"}
<Deadline>      ::=  {<Deadline1>|<Deadline2>|<Deadline3>}
<Deadline1>     ::=  {<Date_pat>{"まで"|"迄"|"中" }("に")?
                     |<Date_pat>"を"{"締"|"〆"}("め")?{"切"|"き"}("り")?}
<Deadline2>     ::=  {"締め切り"|"〆切"|"期限"}{"は" |"が"|"を"}<Date_pat>("まで")?{"と"|"です"|"に"}
<Deadline3>     ::=  {("大")?"至急"|"今すぐ"}
<Contact>       ::=  {"私" |("(")?<Address_pat>(")")?}{"まで"|"宛"}
<Date_pat>      ::=  {({[0-9]+"年"|"平成"[0-9]+"年" })?[0-9]+"月"[0-9]+"日"
                     |[0-9]+"/"[0-9]+("日")?}("(" < 任意の文字 >+")")?< 任意の文字 >*
                     (({"午前"|"午後"})?{[0-9]+"時"[0-9]+"分"|[0-9]+":"[0-9]+})?
<Address_pat>   ::=  {[a-zA-Z0-9_-]+"@"[a-zA-Z0-9_-]+("."[a-zA-Z0-9_-]+)*}
```

(b) 文からの情報抽出に用いるパターン例

図 3 構造と言語のパターンの例
Fig. 3 Expression patterns for structure and language.

3.3.2 ラベル付き箇条書きの判別

ラベル付き箇条書きからスケジュール情報と ToDo 情報を抽出するためには，ラベル付き箇条書きである物理行を特定することが前提となる．箇条書きのラベルには，数字や記号や空白が含まれるなどの種々の特徴がある．この特徴を基にして，ラベル付き箇条書きを特定するためのパターンを構成する．図 3 (a) にパターンの例を示す．ただし，パターン内の数字，アルファベット，ピリオド等の記号や空白は，全角文字と半角文字の双方を許す．

本文または引用文のタグが付けられた各物理行に対して，箇条書きかどうかを判別する手順を**図 4** に示す．まず各物理行の種別を表すタグで囲まれた文字列に対し，パターン ⟨Itemization⟩ を用いてパターンマッチングを行う．パターンマッチングの結果に加え，行の中の空白の有無，ラベルのタイプ，ラベルの文字列の長さとラベルの開始位置を考慮して，各物理行が箇条書きかどうかを判定する．ラベルの文字列は，パターン ⟨Label1⟩ にマッチした文字列から空白を削除した文字列，もしくはパターン ⟨Label2⟩ にマッチした文字列とする．ラベルのタイプは，**表 3** に示すキーワードをラベルの文字列と照合することによって決定する．キーワードを含むラベルのタイプは，そのキーワードが分類されている種別とし，キーワードを含まないラベルのタイプは 'others' とする．ラベルの開始位置は，物理行の文字列の開始位置とラベル内開始位置との和と定義する．ここで，ラベル内開始位置とは，箇条書きラベルの中におけるラベルの文字列の開始位置を表し，パターン ⟨Prefix⟩ にマッチする文字列の長さとする．パターン ⟨Prefix⟩ にマッチする文字列がない場合は，ラベル内開始位置を 0 とする．

箇条書きと判定された物理行に箇条書きのタ

- パターン ⟨Itemization⟩ にマッチするならば
  - 空白を含んでマッチするラベルの候補があるならば
    - ラベルの長さが行全体の長さの 2 分の 1 を上限とする候補を選ぶ
  - マッチしたラベルの文字列に空白がある場合は空白を削除する
  - ラベルの文字列とラベルのタイプを表すキーワードとを照合する
  - ラベルの文字列がキーワードを含むならば
    - ラベルのタイプをキーワードが分類される種別とする
    - 行全体を箇条書きとする
  - そうでなければ
    - ラベルのタイプを 'others' とする
    - ラベルの文字列に空白を含んでいたならば
      - ラベルの文字列がパターン ⟨Suffix⟩ にマッチしたならば，行全体を箇条書きとする ($)
      - そうでなければ
        (1) ラベルの開始位置が，ラベルのタイプが 'others' 以外で文書内に存在する箇条書きのそれに一致するならば，行全体を箇条書きとする
        (2) そうでなければ，箇条書きとしない
    - そうでなければ
      (1) ラベルの文字列がパターン ⟨Prefix⟩ にマッチするならば，行全体を箇条書きとする
      (2) そうでなければ，$にジャンプする
- そうでなければ，箇条書きとしない

図 4 箇条書きの判別の過程
Fig. 4 Process of itemization extraction.

表 3 箇条書きラベルを分類するためのキーワードの例
Table 3 Keywords for classifying itemized labels.

種別	キーワード
date	日時，日程，とき
location	場所，会場，ところ
deadline	締め切り，〆切，期限，期日
contact	連絡先，問い合わせ先，担当

グ ⟨item⟩ を付ける．箇条書きラベルには，ラベルのタイプ，空白を除いたラベルの文字列，ラベル内開始位置，ラベルの文字列の長さをタグ ⟨label type={'date' | 'location' | 'deadline' | 'contact' | 'others'} name='*#name*' l_point='*#point*' length='*#length*'⟩ として付与する．

箇条書きの判別について，図 1 (a) を例にとって述べる．各物理行に対してラベルパターン ⟨Itemization⟩ を用いてパターンマッチングを行う．5 行目の物理行において，"1." がパターン ⟨Prefix⟩ に，"日　　時" が，パターン ⟨Label1⟩ にそれぞれマッチする．マッチした文字列から空白を除いてラベルの文字列 "日時" を得る．ラベルの文字列と表 3 に示すキーワードとを照合し，ラベルのタイプを 'date' とする．ラベルのタイプから 5 行目の物理行を箇条書きと判別する．ラベル内開始位置は，パターン ⟨Prefix⟩ にマッチした "1." から 4 byte とする．6 行目の場所についても同様である．次に 9 行目の物理行において，"3." がパターン ⟨Prefix⟩ に，文字列の長さが 5 byte の "参 加" あるいは 8 byte の "参 加 費" の 2 つの候補がパターン ⟨Label1⟩ にマッチする．9 行目の行全体の文字列の長さである 42 byte の 2 分の 1 を上限として最大の文字列の長さを持つ候補を選ぶので，8 byte の "参 加 費" をラベルとする．ラベルの文字列は，マッチした文字列から空白を削除した "参加費" とし，ラベルのタイプは，ラベルの文字列が表 3 に示すキーワードを含まないので 'others' とする．9 行目の物理行は，ラベルのタイプは 'others' であるが，パターン ⟨Prefix⟩ にマッチしたので，箇条書きと判別する．10 行目の物理行においては，"4." がパターン ⟨Prefix⟩ に，"ハンディ" がパターン ⟨Label2⟩ にそれぞれマッチするので，箇条書きと判別する．12 行目の物理行において，"幹事" がパターン ⟨Label2⟩ に，":" がパターン ⟨Suffix⟩ にともにマッチするので，箇条書きとする．14 行目の物理行においては，"山田" がパターン ⟨Label2⟩ にマッチするが，前後のパターン ⟨Prefix⟩ と ⟨Suffix⟩ にマッチする文字列が存在しないので，文字列の開始位置とラベル内開始位置の和であるラベルの開始位置を比較する．ラベルの開始位置である 0 byte が，ラベルのタイプが 'others' 以外のラベルの開始位置である 4 byte と一致しないので，箇条書きではないと判別する．

3.3.3 箇条書きの範囲の特定と文の切り出し

箇条書きや文章は，複数の物理行にわたることがあるので，その範囲を特定することが必要である．ここで，箇条書きラベルの判別時に付加されたタグ ⟨item⟩ がある物理行を箇条書き行，タグ ⟨item⟩ がない物理行を文章行と定義する．箇条書きと文章の範囲の特定のための条件の分岐を図 5 に示す．箇条書き行と文章行の切替わり，本文か引用文かを示す物理行の種別と文字列の開始位置を用いて，箇条書きまたは文章の範囲を決定する．範囲を決定した後，範囲の先頭を除く各物理行のタグ ⟨own⟩ または ⟨ref⟩ を削除し，範囲の先頭の物理行のタグ ⟨own⟩ または ⟨ref⟩ の属性 p_line に各物理行を埋め込む．箇条書きの場合は，さらに，箇条書きの範囲を囲むようにタグ ⟨item⟩ を付加する位置を修正する．以上の処理の後に，各文章の範囲に対して，句点等による文の切出しを行う．切り出された文ごとにタグ ⟨sentence⟩ を付ける．最後に，タグ

- n 行目が箇条書き行ならば
  - 変数 k を 1 にセットする
  - $n+k$ 行目の物理行の種別が n 行目と同じで，かつ，$n+k$ 行目が文章行ならば ($)
    * n 行目が箇条書きラベルのみならば
      (1) $n+k$ 行目の開始位置が n 行目の開始位置と等しいまたは大きいならば，k をインクリメントし，$にジャンプする
      (2) そうでなければ，n 行目から $n+k-1$ 行目までを箇条書きの範囲とする
    * そうでなければ
      (1) $n+k$ 行目の開始位置が n 行目の開始位置よりも大きいならば，k をインクリメントし，$にジャンプする
      (2) そうでなければ，n 行目から $n+k-1$ 行目までを箇条書きの範囲とする
  - そうでなければ，n 行目から $n+k-1$ 行目までを箇条書きの範囲とする
- そうでなければ
  - 変数 k を 1 にセットする
  - $n+k$ 行目の物理行の種別が n 行目と同じならば ($$)
    (1) $n+k$ 行目が文章行ならば，k をインクリメントし，$$にジャンプする
    (2) そうでなければ，n 行目から $n+k-1$ 行目までを文章の範囲とする
  - そうでなければ，n 行目から $n+k-1$ 行目までを文章の範囲とする

図 5 箇条書きの範囲の特定と行の接続の過程
Fig. 5 Process of specifying scopes of itemizations and connecting sentences.

⟨item⟩ と ⟨sentence⟩ の属性 l_line に，これらのタグの先頭からの順番を論理行番号として割り当てる．

タグ付けによって図 1 に示す電子メール文書の文書構造の認識を行った結果を**図 6** に示す．なお，文書の本体はタグ ⟨body⟩ で囲まれる．

3.4 情報抽出

3.4.1 ラベル付き箇条書きからの情報抽出

箇条書きのタグ ⟨item⟩ が付けられた範囲から，情報を抽出する．箇条書きラベルのタグ ⟨label⟩ の属性 type に応じて，以下のように情報を抽出する．

- date と deadline
  タグ ⟨item⟩ で囲まれた範囲の中で，タグ ⟨label⟩ で囲まれた範囲を除いた文字列から，図 3 (b) に示すパターン ⟨Date_pat⟩ を用いたパターンマッチングにより抽出された日時情報
- location と contact
  タグ ⟨item⟩ で囲まれた範囲の中で，タグ ⟨label⟩ で囲まれた範囲を除いた文字列のすべて
- others
  タグ ⟨item⟩ で囲まれた範囲の中で，タグ ⟨label⟩ の属性 name に格納されたラベルの文字列と，タグ ⟨label⟩ で囲まれた範囲を除いた文字列のすべて

```
<body>
<own p_line='1 2 3' s_point='0'>
<sentence l_line='1'>第３回小松部長杯ゴルフコンペを下記のとおり開催いたします．</sentence>
<sentence l_line='2'>みなさま奮ってご参加いただきますようよろしくお願いいたします．</sentence>
<sentence l_line='3'>なお，詳細は別途連絡いたします．</sentence>
</own>
<own p_line='5 6' s_point='0'>
<item l_line='4'><label type='date' name='日時' l_point='4' length='8'>1．日　時</label>平成１０年６月８日（月） エメラルドコース ８：００スタート</item>
</own>
<own p_line='7 8' s_point='0'>
<item l_line='5'><label type='location' name='場所' l_point='4' length='8'>2．場　所</label>葉山国際カンツリー倶楽部 [三浦群葉山町上山口 Tel0468-12-1234] </item>
</own>
<own p_line='9' s_point='0'>
<item l_line='6'><label type='others' name='参加費' l_point='4' length='8'>3．参 加 費</label>約２５，０００円（昼食別）</item>
</own>
<own p_line='10' s_point='0'>
<item l_line='7'><label type='others' name='ハンディ' l_point='4' length='8'>4．ハンディ</label>新ペリア方式にて算出</item>
</own>
<own p_line='12' s_point='0'>
<item l_line='8'><label type='others' name='幹事' l_point='0' length='4'>幹事:</label>ご質問等は山田 (yamada@abc.co.jp) まで</item>
</own>
<own p_line='14' s_point='0'>
<sentence l_line='9'>山田 太郎 (yamada@abc.co.jp)
</sentence>
</own>
</body>
```

(a) 電子メール文書例 1 の文書構造

```
<body>
<own p_line='1 2 3 4 5 6 7 8 9' s_point='0'>
<sentence l_line='1'>大原です．</sentence>
<sentence l_line='2'>１２月１０日（水）１５：００～１６：３０に１０４Ｃ会議室で前回未受講の社員を対象とした電子決裁システム講習会が行われます．</sentence>
<sentence l_line='3'>未受講の人は，やむを得ない事情がない限り出席せよとのことです．</sentence>
<sentence l_line='4'>各位受講実績と参加の可否（否の場合は理由を添えて）を大原まで１１／１８日中にご回答ください．</sentence>
<sentence l_line='5'>よろしくお願いします．</sentence>
</own>
</body>
```

(b) 電子メール文書例 2 の文書構造

図 6 文書構造のタグが付与された電子メール文書の本体
Fig. 6 Bodies of e-mail messages with tags for message structure.

3.4.2 文章からの情報抽出

文章として記述されたスケジュール情報と ToDo 情報を抽出するために，タグ ⟨sentence⟩ が付けられた各文に対してパターンマッチングを行う．パターンマッチングに適用するためのパターンは，実際に我々が受信した電子メール文書から得られた言語の特徴に基づいて構成される．図 3 (b) にパターンの例を示す．主張タイプの発話行為の表現をパターン ⟨Inform⟩ に，依頼タイプの発話行為の表現をパターン ⟨Request⟩ にそれぞれ取り入れる．

パターンマッチング処理の原則を下記に示す．

- パターンが複数からなる場合は，前にある方からパターンマッチングをそれぞれ行う．
- 文に1つのパターンがマッチした場合，残っているパターンでパターンマッチングを行わない．
- ⟨$Location+⟩ や ⟨$Event+⟩ は，1つ以上の連続する任意の文字にマッチするとする．
- パターンマッチングは，パターン ⟨Location_rep⟩ を除いて最長マッチングとする．
- 任意の文字列のパターンが読点を含む文字列にマッチングした場合，読点より前の文字列は削除する．

図1(b)にタグを付けた図6(b)を例にとり文章中からの情報抽出について述べる．タグ ⟨sentence⟩ の付けられた各文に対して，図3(b)に示すパターン ⟨Schedule⟩ と ⟨ToDo⟩ を用いてパターンマッチングを行う．論理行が2行目の文が，主張タイプの発話行為を含むパターン ⟨Schedule⟩ にマッチし，“１２月１０日（火）１５：００”を開始日時とし，“１６：３０”を終了日時とし，“１０４Ｃ会議室”を場所とし，“電子決裁システム講習会”をイベント名として，それぞれを抽出する．論理行が4行目の文が，依頼タイプの発話行為を含むパターン ⟨ToDo⟩ にマッチし，“１１／１８日”を期限として抽出する．

3.4.3 情報の正規化

電子メール文書の本文のみでは，完全な日時や期限の情報を得られない場合が少なからずある．たとえば，文書を書く人や書く状況によっては，年や月は省略するなど開始日が完全でないことがある．また，その日1日で終了するスケジュール情報の通知には，通常終了日は明示的に書かれていない．さらに，絶対的な日時の表現だけでなく，“明日”や“来週の月曜日”等の相対的な日時の表現が用いられる場合もある．

PIM ソフトウェアへの入力等の応用を考えた場合，完全な日時情報を得られないことは都合が悪い．このような場合は，以下のルールを用いることによって，情報を補完する．

- 年や月の情報がない場合は，送信日時を表すメールヘッダの Date フィールドから補完する．
- 相対的な日時表現は，ヘッダ情報にある Date フィールドを基準として，カレンダーに基づく日時と曜日の演算により，絶対的な表現に正規化する．
- 終了日がない場合は，開始日と同一日で補完する．

3.5 複数の抽出情報の関連付け

スケジュールや ToDo は，1つの文書に複数個存在する場合がある．これらの情報の個数を，スケジュールは開始日の個数，ToDo は依頼タイプの発話行為の個数を基準に定義する．スケジュールや ToDo が複数個存在する場合には，抽出された複数のスケジュール情報や ToDo 情報をそれぞれ正しく関連付けることが必要である．そこで，

- 箇条書きと文章から抽出されたそれぞれの開始日時の比較による情報のマージ，
- 基準とする情報とそれ以外の情報の論理行番号の最短距離による関連付け，

というルールを用いて，抽出された複数のスケジュール情報と ToDo 情報の関連付けを行う．スケジュールや ToDo の個数を定義する基準として用いる開始日や依頼タイプの発話行為の情報の個数に対して，これらの情報に関連付ける情報の個数が足りない場合は，同種の情報の中で論理行番号の距離が最も近い情報を用いて補完する．

本文からスケジュール情報の開始日が抽出されなかった場合は，引用文からの抽出を行う．ToDo 情報については，本文にしか存在しえないので引用文からの抽出は行わない．ただし，引用文中に期限が記述されている場合は，付随情報としての価値があるので期限の情報を抽出する．抽出方法は，本文からの抽出方法と同様である．

4. 評価実験

本稿で提案したアルゴリズムを Perl を用いて計算機上に実装した．図1(b)に示す文書を入力としたときの出力結果を**図7**に示す．出力形式は，インターネットを介してスケジュールや ToDo を送受信する規格である vCalendar [11] に準拠している．ただし，2バイト文字は本来エンコードされるが，例示するためにエンコードしていない．VEVENT の SUMMARY はイベント名を，DTSTART と DTEND は開始日時と終了日時を，LOCATION は場所をそれぞれ表す．VTODO の DUE は期限を表す．DTSTART，DTEND および DUE では，多様な日時表現は正規化され，数値のみの日時表現が用いられる．VEVENT と VTODO の UID はそれぞれが持つ固有の ID を，RELATED-TO は UID に関連する ID を表し，E と T の後にシーケンシャルな番号とメールヘッダの Message-Id を付けることによって記述される．本例では期限の時刻は記述されていないので，デフォルトとして 235900 としている．なお，VTODO の SUMMARY は一律 ‘reply’ としている．

電子メール文書からの情報抽出についての評価実験

```
BEGIN:VCALENDAR
VERSION:1.0
BEGIN:VEVENT
SUMMARY;LANGUAGE=ja-JP;CHARSET=ISO-2022-JP;
ENCODING=BASE64:電子決裁システム講習会
DTSTART:19971210T150000
DTEND:19971210T163000
LOCATION;LANGUAGE=ja-JP;CHARSET=ISO-2022-JP;
ENCODING=BASE64:１０４Ｃ会議室
UID:E1<199711120122.KAA04114@abc.co.jp>
RELATED-TO:T1<199711120122.KAA04114@abc.co.jp>
END:VEVENT
BEGIN:VTODO
SUMMARY:reply
DUE:19971118T235900
UID:T1<199711120122.KAA04114@abc.co.jp>
RELATED-TO:E1<199711120122.KAA04114@abc.co.jp>
END:VTODO
END:VCALENDAR
```

図 7 スケジュールおよび ToDo の抽出処理の出力結果
Fig. 7 Output from schedule and to-do extraction.

にあたって，実験の条件は次のとおりである．文書構造を認識するためのパターンの数は 14，パターンに用いた文字列の数は 62 で，情報抽出のためのパターンの数は 141，パターンに用いた文字列の数は 964 である．文字列には数字と記号も含んでいる．パターンの数と文字列の数ともに，省略可能なものについては省略した場合と省略しない場合を別々に数えている．対象データは，アルゴリズムを規定するために調べた電子メール文書 80 通（既知データ）と実際に我々が受信した電子メール文書 450 通（未知データ）である．

実験結果を評価するにあたって，以下の 2 つの基準に分けて評価を行う．

(1) スケジュールと ToDo のフィルタリングの精度
(2) スケジュール情報と ToDo 情報の抽出の精度

前者の精度については，スケジュールと ToDo を含む電子メール文書の検出における再現率と適合率で表す．スケジュールのフィルタリングは，文書中に何らかのアポイントメントやイベントの日時と場所を含むことと定義し，ToDo のフィルタリングは，文書中に返信を要求する依頼タイプの発話行為を含むことと定義した．日時は開始日，終了日，開始時刻，終了時刻から構成され，少なくとも開始日を含むとした．後者の精度については，スケジュールと ToDo を構成する情報の抽出における再現率と適合率で表す．表中の項目の“その他”とは，日時，場所，期限，連絡先以外の箇条書きで記述された付随の情報をまとめたものである．

正解かどうかの判断は，それぞれ次の情報を用いることによって行った．開始日，終了日，開始時刻，終了時刻，期限の日時表現については，メールヘッダからの情報の補完と相対表現から絶対表現への正規化による日時情報を用いた．場所については，文章中からの場合は抽出された文字列を，箇条書きからの場合は箇条書きの内容すべてを用いた．連絡先については，文章中からは電子メールアドレスを，箇条書きからは箇条書きの内容すべてを用いた．文章中から抽出する場合，電子メールアドレスがない場合は該当なしとした．イベント名については，抽出された文字列が中心的な情報を含み，文字列自身で意味をなしている場合を正解とした．その他の箇条書きは，ラベルとその内容の中心的な部分が抽出されていれば正解とした．ネストされた箇条書きについては，トップの箇条書きの数を該当数とし，ネストされた箇条書きはトップの箇条書きの内容とした．

5. 考　　察

5.1 評　　価

目的や形式が異なりバラエティに富む電子メールの実文書を対象に抽出実験を行った結果について，スケジュールと ToDo のフィルタリングの精度と，スケジュール情報と ToDo 情報の抽出の精度とをあわせて**表 4** に示す．表 4 は大きく左右に分かれ，それぞれ既知データと未知データに対する実験結果を示している．実験の結果，既知データと未知データのいずれも，スケジュールと ToDo のフィルタリングと情報抽出において，再現率適合率ともに 90%前後という良好な結果を得ることができた．実用に十分に耐える程度の精度が得られたことから，本手法の有効性を確かめることができた．

スケジュールと ToDo のフィルタリングについては，スケジュールの方が ToDo よりも再現率が低い．これは，スケジュールのフィルタリングの定義はイベントの開始日と場所の 2 つを含まなければならないのに対し，ToDo のフィルタリングの定義は依頼タイプの発話行為のみを含めばよいとしたことによる．そのうえさらに，日時の表現や場所の表現は，表記のゆれや誤りを多く含んでいるために，正確な情報抽出が難しいからである．

スケジュール情報と ToDo 情報の抽出については，**表 5** に抽出に失敗した原因別頻度分布を示す．項目として，日時，場所，イベント名，期限，連絡先をあげた．ただし，日時は開始日，終了日，開始時刻，終了時刻のすべてをまとめてある．これらの項目について，それぞれ既知データと未知データについて調べた．失敗の原因として，“箇条書きラベルがない”，“発話行為がない”，“パターンのアンマッチ”，“意図しない

表 4 スケジュールと ToDo のフィルタリングと情報抽出の精度
Table 4 Recall and precision of schedule and to-do extraction.

項目	既知データ			未知データ		
	該当数	正解数 (再現率)	抽出数 (適合率)	該当数	正解数 (再現率)	抽出数 (適合率)
スケジュール	65	60 (92.3%)	60 (100 %)	119	105 (88.2%)	107 (98.1%)
開始日	75	69 (92.0%)	69 (100 %)	135	119 (88.1%)	119 (100 %)
終了日	10	10 (100 %)	10 (100 %)	14	14 (100 %)	14 (100 %)
開始時刻	65	59 (90.8%)	59 (100 %)	108	95 (88.0%)	95 (100 %)
終了時刻	32	29 (90.6%)	29 (100 %)	79	70 (88.6%)	73 (95.9%)
場所	67	59 (88.1%)	59 (100 %)	115	99 (86.1%)	101 (98.0%)
イベント名	72	64 (88.9%)	67 (95.5%)	116	100 (86.2%)	105 (95.2%)
その他	127	118 (92.9%)	127 (92.9%)	401	368 (91.8%)	391 (94.1%)
ToDo	51	50 (98.0%)	51 (98.0%)	173	168 (97.1%)	172 (97.7%)
期限	29	29 (100 %)	29 (100 %)	100	91 (91.0%)	92 (98.9%)
連絡先	21	19 (90.5%)	19 (100 %)	53	44 (83.0%)	44 (100 %)

表 5 抽出失敗原因の頻度分布
Table 5 Causes of extraction errors and failures.

失敗原因	日時		場所		イベント名		期限		連絡先	
	既知	未知	既知	未知	既知	未知	既知	未知	既知	未知
ラベルなし	3	6	3	9				1	1	1
発話行為なし	2	2	2	2	2	2				
アンマッチ		9		2		5		7		1
意図せぬマッチ		2	3	2	4	5		1		1
関連付けミス					1	1		1		
知識を必要	1	2		2	1	3			1	6

マッチ", "抽出情報の関連付けの誤り", "知識を必要とする表現" の 6 つに分類した．特徴としていえることは，期限の抽出で未知データに "パターンのアンマッチ" が大きな割合を占めたことと，場所の抽出で "箇条書きラベルがない" が大きな割合を占めたことである．しかし，パターンのアンマッチは，既知データにないパターンが未知データにあったことを示すので，この問題はパターンの種類を増やすことにより解決できる．場所の "箇条書きラベルがない" の代表的な例として，日時の箇条書きラベルの中に場所の情報が併記されているケースがあげられる．連絡先については，知識を必要とする表現が最も多かった．代表的な例として，"下記に連絡下さい" のような表現の直後に単独で電子メールアドレスが記述されているケースがあげられる．

情報抽出の精度をさらに上げるためには，パターンのアンマッチが失敗の大きな原因なので，多くのパターンを用意することが最も有効であると考えられる．ところが，パターンを追加すると一般的には誤ってマッチする副作用が発生する頻度も増加する．評価実験でパターンのアンマッチが特に多かったのは日時と期限とイベント名である．日時と期限のパターンは，いずれも数字や記号などの文字の種別と言語表現を同時に含んでいる．このため，パターンを追加しても，文字の種別と言語表現の両方を同時に誤ってマッチするケースは多くはないと考えることができ，副作用は少ないと考えられる．逆に，イベント名のパターンは文字の種別に頼らないため，日時や期限に比べると誤ってマッチする副作用は多いと予想される．そのため，イベント名にも文字の種別の制約を課したり，同一文における複数のイベント名の候補に対する選択の基準を設けたりすることが必要である．たとえば，文字の種別の制約としては，少なくとも漢字やカタカナを 1 文字含む等が考えられる．また，イベント名の選択の基準としては，発話行為の表現ごとに指定したパターンの優先順序や，文内におけるイベント名と発話行為の表現の位置関係等が考えられる．パターンの追加以外にも，精度を上げるためには，箇条書きラベルが存在しない場合や照応表現が用いられる場合に対処するためのルールを設けることも有効な方法である．また，頻度は少ないものの，日時や場所を表す箇条書きラベルが列のように横に並ぶ表形式に対し，それぞれの情報を抽出することができなかった．このような表形式に対処するレイアウト情報を考慮することも有効な方法である．

5.2 議　　論

本稿で提案した手法は，箇条書きを認識する過程において，スタイル情報だけでなく，箇条書き行の文字

列の長さや位置の情報を考慮するため，箇条書きラベルが分かち書きされている場合や箇条書きが複数行にわたる場合でも内容の抽出が可能である．また，箇条書きラベルをタイプ別に分類するためのキーワードの種別は追加変更が容易なため，文書構造の認識はドメインに依存しない．パターンマッチングにより文章からの情報を抽出する過程では，形態素解析の結果やドメインに依存する特定のキーワードは用いず，普遍的な日時の表現と発話行為の表現を含む言語の特徴に着目したパターンを用いている．このため，未知語や誤字脱字による影響を受けずに，ドメインに依存しないスケジュール情報やToDo情報を文章から抽出することができる．物理的な文書構造の種別と抽出した情報の存在位置を用いた情報の関連付けにより，電子メール文書に複数のスケジュールやToDoが含まれる場合でも，これらの情報を抽出することが可能である．

今後の課題は，ユーザが必要に応じてパターンマッチングをカスタマイズするための仕組みを設けることである．なぜならば，各々のユーザが受信する電子メールには，ユーザごとに異なる特定の言語表現が含まれることが考えられるからである．パターンマッチングのカスタマイズを行うには，ユーザが視覚的に分かりやすくパターンを記述できるツールが求められる．さらに，ユーザが新たに記述したパターンの有効性を確認できるように，パターン記述のツールにはパターンの無矛盾性や曖昧性を保障する機能が必要である．

6. おわりに

本稿では，ユーザが受信する電子メールにおいて，電子メールの整理やPersonal Information Manager（PIM）ソフトウェアとの連携を実現するために，電子メールからスケジュール情報とToDo情報を抽出することを目標とした．電子メールは任意の目的や任意の形式を許すため，スケジュール情報やToDo情報は様々な文書構造や言語表現により伝達される．本稿では，電子メール文書の構造と言語の特徴に着目し，それに基づくパターンマッチングを用いた文書構造の認識と情報抽出および文書構造の種別と抽出した情報の存在位置を用いた情報の関連付けの手法を提案した．本手法は，ユーザが受信した電子メール文書を入力とし，スケジュールとToDoを含む文書をフィルタリングし，スケジュール情報とToDo情報を抽出して汎用の形式で出力することが可能である．電子メールの実文書を対象とした情報抽出の実験結果は，スケジュールとToDoを含む電子メール文書のフィルタリングや，PIMソフトウェアへの入力等の実用に耐えうる高い精度を有することを実証した．これにより，電子メール文書を取捨選択したり個人情報を管理したりするユーザの負荷が大幅に軽減されることが期待できる．

今後は，照応表現や様々なレイアウト情報が用いられた情報を抽出する手法や，ユーザによるパターンマッチングのカスタマイズのためのパターン記述を支援する手法の研究に取り組んでいく．

謝辞 本稿の執筆にあたり，貴重なコメントをいただきました，NTTサイバースペース研究所の小池秀幸氏，小原永氏ならびに永田昌明氏をはじめとするみなさまに慎んで感謝の意を表します．

参考文献

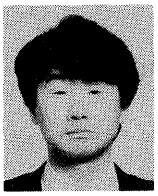
1) Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D. and Tyson, M.: FASTUS: A Finite-state Processor for Information Extraction from Real-world Text, *Proc. 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pp.1172–1178 (1993).
2) Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D., Kameyama, M., Kehler, A., Martin, D., Myers, K. and Tyson, M.: SRI International FASTUS System MUC-6 Test Results and Analysis, *Proc. 6th Message Understanding Conference (MUC-6)*, pp.237–248 (1995).
3) 江里口善生，木谷 強：パターンマッチング手法による名称特定処理の有効性の検討，情報処理学会研究報告，NL115-10, pp.67–73 (1996).
4) 原 正巳，木谷 強，江里口善生：特徴的表現を利用した特許抄録作成法の検討，情報処理学会研究報告，NL100-14, pp.105–112 (1994).
5) 稲垣博人：事象解析による要約情報の抽出，情報処理学会研究報告，NL84-3, pp.17–24 (1991).
6) 小松英二，加藤安彦，安原 宏，椎野 努：要約支援システムCOGITO—文書の構造解析，情報処理学会研究報告，NL64-11, pp.85–91 (1987).
7) 松尾比呂志，木本晴夫：抽出パターンの階層的照合に基づく日本語テキストからの内容抽出法，情報処理学会論文誌，Vol.36, No.8, pp.1838–1844 (1995).
8) *Proc. 6th Message Understanding Conference (MUC-6)*, Morgan Kaufmann (1995).
9) 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, pp.2371–2379 (1995).
10) 高松 忍，西田富士夫：見出し情報を用いたテキスト解析と情報抽出，情報処理学会論文誌，Vol.29, No.8, pp.760–769 (1988).
11) vCalendar: The Personal Calendaring and Scheduling Exchange Format: A versit Consortium White Paper, http://www.imc.org/pdi/vcalwhite.html (1997).

12) Watanabe, H.: A Method for Abstracting Newspaper Articles by Using Surface Clues, *Proc. 16th International Conference on Computational Linguistics* (*COLING-96*), pp.974–979 (1996).
13) 山梨正明：発話行為，新英文法選書 12，大修館書店 (1986).

(平成 10 年 11 月 2 日受付)
(平成 11 年 9 月 2 日採録)

長谷川隆明（正会員）

1969 年生．1992 年慶應義塾大学理工学部電気工学科卒業．1994 年同大学大学院理工学研究科計算機科学専攻修士課程修了．同年，日本電信電話（株）入社．現在，NTT サイバースペース研究所メディア処理プロジェクトに勤務．自然言語処理，情報抽出，エージェントアーキテクチャに関する研究に従事．日本ソフトウェア科学会会員．



高木伸一郎（正会員）

1979 年金沢大学工学部電気工学科卒業．1981 年同大学大学院電気工学専攻修士課程修了．同年，日本電信電話公社（現 NTT）横須賀電気通信研究所入所．日本語形態素解析を応用した日本文校正支援システムの研究開発を経て，合成音声による日本文読み上げ技術とテキストマイニング技術を応用した知的支援サービスの開発に従事．現在，NTT サイバースペース研究所メディア処理プロジェクト主幹研究員．電子情報通信学会会員．