自然言語処理　132－4
(1999. 7. 22)

表層情報を利用したネットニュース領域構造解析

浅野 久子　　永田 昌明

NTT サイバースペース研究所
〒239-0847 神奈川県横須賀市光の丘 1-1
{hisako,nagata}@nttnly.isl.ntt.co.jp

あらまし

インターネットを流通する電子化情報のうち、構造化が行われておらず、引用を表す記号等、通常のテキストとは異なる文字の用法が存在するネットニュースや電子メールを対象として、情報抽出や要約を容易にするための自動構造解析を検討している。このうち本稿では、引用構造、および、内容的な領域区分 — ある投稿者が記述した文章、ニュースリーダが自動的に挿入した文章、署名 — を、空行等により分割された領域単位に、表層的に得られる文字種や位置情報等を属性とした決定木を用いて解析する方法を提案する。また、ネットニュースコーパスを用いた実験を行い、本手法の有効性を示す。

キーワード　領域構造解析, ネットニュース, XML, C4.5, 決定木

NetNews Area Analysis Using Surface Information

Hisako Asano　　Masaaki Nagata

NTT CyberSpace Laboratories
1-1 Hikari-no-oka Yokosuka-shi Kanagawa 239-0847 Japan
{hisako,nagata}@nttnly.isl.ntt.co.jp

Abstract

For information extraction and summarization of NetNews and e-mails, it is necessary that the structure of these electronic texts are analyzed previously because they are not structured and exist extra character usage such as quotation marks (e.g. "> "). We propose a method which analyzes quotation structure and area classification by contents — text by sender, text by news-reader and signature — using decision trees which have surface information as properties. Experiments using our NetNews corpus shows that this method is effective.

key words　Area structure analysis, NetNews, XML, C4.5, Decision tree

1 はじめに

近年、インターネットの急速な発展とともに、WWWやネットニュース、電子メールなど、多くの電子化情報が流通するようになり、これらの膨大な情報を処理するために、情報抽出や要約、重要文抽出などの技術が注目を集めている。このうち、WWWのHTML文書は、HTMLにより情報の構造化が行われている。しかしネットニュースや電子メールは、通常は構造化されておらず、また、文字で描いた図や引用を表す記号等、通常のテキストとは異なる文字の用法が存在し、改行位置や句読点等の表現も多様である。このため、重要文抽出などの技術の基盤となる言語処理の基本単位(例えば「文」など)を把握するのも難しい。

我々は、これらのテキストを対象とした住所録情報[1]やスケジュール情報[3]などの情報抽出や要約等を容易にするための構造化を目標として、以下の3つの観点に基づいたネットニュース構造化用XMLタグセットを提案している[2]。

処理区分領域の設定 ネットニュースのボディは、書き手という観点では、投稿者自身が書いた領域と、他の記事を引用した領域(複数の記事を並列に引用したり、多重的に引用することもある)に分けられる。また内容的には、ある投稿者が書いた文章(通常文章)のほかに、ニュースリーダによって自動的に挿入された文章(自動挿入文章、例:"Taroh Yamada said ...")、署名(signature)、文字で描いた図、添付書類などの領域がある。情報抽出・要約では、これらのある特定の領域を処理対象とすると考えられるため[1]、この処理を区分すると考えられる単位である、引用、通常文章、自動挿入文章、signature、図、添付書類という領域に分割する。このうち、引用はすべての処理区分領域を再帰的に内包しうる。

処理単位の設定 情報抽出・要約では、通常、段落や文などの言語単位を処理単位とするため、これらの処理単位の設定を行う。

[1]例えば重要文の抽出では、最も重要な文はその記事の投稿者が書いた文章の中にあると考えられるので、引用内の文章よりも重みづけする、signatureや図などは対象としないなど。

表1: ネットニュースコーパス

ニュースグループ	記事数
fj.os.ms-windows	578
fj.fleamarket.ticket	583
fj.life.health	280

元テキストへの再現性 ネットニュースでは、文字による装飾や、空白文字の挿入によるセンタリング等のレイアウト的表現により強調などが行われる場合がある[5]ため、これらの文字を保存することが必要となる。しかし、言語解析においては、解析誤りなどの原因になる等の悪影響を及ぼすので言語的に意味をもつ文字と区別できるようにする。

さらに我々は、特徴の異なる3つのニュースグループの97年8～9月の記事を対象に、[2]のXMLタグセットを付与したネットニュースコーパスを作成している(表1)。

本稿では、[2]で示したネットニュース構造化のうち、処理区分領域を自動的に設定する方法を提案する。主な特徴は、表1のコーパスを学習データとして、表層的に得られる情報を属性としてC4.5[4]により生成された決定木を用いて、空行等で分割されるブロックという単位で領域判定を行うことである。ここで引用領域は、すべての処理区分領域を再帰的に内包しうるので、まず、引用領域の構造を解析した後、各引用構造内を対象に、他の処理区分領域(これを以後内容種別とよぶ)を解析する。例えば、図1では、図の左側に示すような引用構造を設定した後に、点線で区切られた領域別に、図右側に示したように内容種別の設定を行う。

2 引用構造の解析

2.1 処理の単位と流れ

引用領域の特徴として、

1. 引用領域は、行頭に引用を表す記号類(これを引用記号とよぶ)をつけることによって表される。

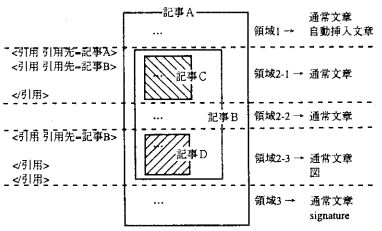

図 1: ネットニュースの構造例

2. 同一記事からの引用には、同一の引用記号が用いられる。

3. 階層的な引用構造は、引用記号の入れ子により表される。

という傾向が非常に強いことが挙げられる。そこで、引用構造解析を、以下の手順により設定される“引用判定ブロック”を単位とすることとした。

1. ボディを空行により分割する。（前述の特徴1より、空行は引用領域にならないと考えられる。）

2. 記事中の任意の連続する複数行において、行頭が同一文字列である場合、これを引用記号候補列として抽出する（引用記号候補列種類＝連続行）。ただし、引用記号候補列の末尾文字に、ひらがな、カタカナ、漢字、英数字は認めない。また、空白文字のみからなる引用記号候補列も認めない。さらに、行頭が“0文字以上4文字以下の‘ ’(スペース)”+“1文字以上3文字以下の‘>’”+“0文字以上4文字以下の‘ ’(スペース)” [2]である場合も引用記号候補列とする（引用記号候補列種類＝単独行）。また、引用構造を階層的に扱うために、ある引用記号候補列を内包する引用記号候補列は削除する（例：“> >”と“> ”が抽出された場合、“> >”を削除）。

3. 1. で分割された各領域において、引用記号候補列を含む行が存在する場合、同一の引用記号候補列が途切れる行間でさらに分割する。

[2]引用記号として最も一般的な文字列であり、不連続に出現した場合でも、引用記号である可能性が高いため。

```
1   Yamada Taro wrote:
2   | > という2行に分割されます。
3   |
4   | 確認しました。そういうことでしたか。
5   | 私の場合、特定の一人からのメールのみそうなり
6   ますが、相手の
7   | メーラーも原因なのでしょうか？
8
9   いえいえそうじゃありません。我々が細々と使って
10  いるこの Communicator もしっかり分割して
11  いますよ。
```

図 2: ネットニュース記事ボディ例

また、1行の文字列が一定の長さを越えたときに、ニュースリーダが自動的に改行を挿入する自動改行行に対応するため、ある1行が同一引用記号候補列を含む行に挟まれている場合には、その行をその前後の行と同一ブロックとして扱う。

そして、この引用判定ブロック単位に2.2節の方法を用いて引用であるか判定した後、引用と判定された全ブロックを引用記号（＝引用と判定されたブロックの引用記号候補列）別のグループに分け、グループ別に、行頭から引用記号を削除した文字列に対して再度引用判定ブロックを設定し、引用判定を行う。この再帰的な引用判定は、再設定されたすべてのブロックが“引用領域”ではないと判定されるまで繰り返される。

例えば、図2の記事では、“| ”が引用記号候補列となり、第1行、第2～7行、第9～11行という3つの引用判定ブロックに分割して引用領域判定判定を行う。ここで第2～7行が引用領域と判定された場合、このブロックから引用記号“| ”を取り除いた文字列を対象にブロックの再設定を行い、“> ”を引用記号候補列として、第2行、第4～7行の2ブロックを設定し、再度引用領域判定を行う。

2.2 引用領域判定

各引用判定ブロックが引用であるかを判定するために、決定木を利用する。引用判定ブロックは、単独の領域からなるとは限らないので、“引用領域”、“引用領域以外”、“引用領域と引用領域以外の混合”の3クラスに分類する。判定に用いる13属性とその取り得る値を表2に示す。属性は、

表 2: 引用領域判定に用いる属性

No.	属性	値	備考
1	ブロック行数	連続値	ブロックの行数
2	ブロック末尾文字種	43 種	全半角別, 記号を細分類
3	正順ブロック位置	連続値	先頭ブロックからの位置
4	逆順ブロック位置	連続値	末尾ブロックからの位置
5	直前空行行数	連続値	ブロック直前の空行数
6	直後空行行数	連続値	ブロック直後の空行数
7	平均行頭空白文字行 / ブロック	連続値	行頭 (*1) が空白の行数 / ブロック行数
8	平均空白分割数 / 行	連続値	空白 or 改行で分割された文字列数 (*1)/ ブロック行数
9	引用記号候補列長	連続値	バイト長
10	引用記号候補列先頭文字種	43 種, なし	全半角別, 記号を細分類
11	引用記号候補列末尾文字種	43 種, なし	全半角別, 記号を細分類
12	引用記号候補列準末尾文字種	42 種, なし	末尾に最も近い半角空白以外の文字種
13	引用記号候補列種類	なし, 連続行, 単独行	2.1節手順 2 参照

(*1) 引用記号候補列は除く

1 ブロックのレイアウト的な特徴を表す情報（属性 No.1 ～ 8）と、引用記号候補列の特徴を表す情報（属性 No.9 ～ 13）という観点から選択した。

ここで、文字種は、漢字、カタカナ、ひらがな、および、半角と全角を区別した英字、数字、記号類（36 種に細分類）のいずれかの値をとる。記号を細分類したのは、'>'（引用記号）や'#'（コメント行）や'・'（箇条書き）のように、ある特定の意味を表しやすい文字を区別して扱うためである。この記号の細分類の効果は、2.3節で検証する。

2.3 実験

提案した引用領域判定手法の評価を行うため、表 1のコーパスを、表 3のように、訓練データ記事（約 9 割）とテストデータ記事（約 1 割）に分け、訓練データを用いて C4.5 により決定木を作成した。ここでは、引用内の再帰的なデータは用いなかった。

2.3.1 作成された決定木の特徴

表 3の訓練データ（8289 ブロック）により作成された決定木の特徴は、以下の 2 点であった。

- 訓練データ中に "引用領域と引用領域以外の混合" クラスとなる引用判定ブロックは 2 つしかなく、その結果、決定木に "引用領域と引用領域以外の混合" クラスとなるリーフは存在しなかった。これは、2.1節で設定した処理単位が適切であったことを示している。

表 3: 訓練データとテストデータ

ニュースグループ	訓練データ	テストデータ
fj.os.ms-windows	522	56
fj.fleamarket.ticket	523	60
fj.life.health	253	27

- 決定木のノードとして利用された属性は、No.3, 9, 10, 11, 12 の 5 種類のみであった。これより、ブロックのレイアウト的な特徴を表す情報はあまり重要ではなく、引用記号候補列の特徴を表す情報が重要であることがわかる。

2.3.2 判定精度

表 4に、提案手法の訓練データ（再帰解析なし）と準訓練データ（再帰解析あり）[3]とテストデータ（再帰解析あり）に対する判定精度、および、比較対象として、引用記号として最もよく用いられる '>' が行頭から 10 バイト以内に存在した場合に、その行を引用領域とした場合の誤り率を示す。誤り率は、引用判定ブロックおよび行を単位として算出した。ここで、決定木手法の訓練データと ">= 引用" 手法では再帰解析を行っていないため、第一段階の引用の有無のみを評価対象とした。決定木手法の準訓練データとテストデータでは、全階層における引用の有無を評価対象とし、

[3]訓練データに対し、引用領域の再帰的な解析も行ったものであり、再帰解析部分は未訓練データとなる。

表 4: 引用領域判定精度

手法	決定木				'>'=引用
	記号 36 種	記号 4 種	記号 36 種	記号 36 種	
データ	訓練データ		準訓練データ	テストデータ	全データ
全ブロック数	8289		10144	1167	-
全行数	24145			2647	26792
誤りブロック数	11	67	14	7	-
ブロック誤り率	0.13%	0.81%	0.14%	0.60%	-
誤り行数	64	208	71	14	547
行誤り率	0.27%	0.86%	0.29%	0.53%	2.04%

行を単位とした評価では、全階層が正しく判定されているかどうかを基準とした。

訓練データについては、文字種の分類として、記号 36 分類をスペース、タブ、全角記号、半角記号の 4 種に減らした場合も示す。

2.3.3 考察

表 4より、本手法は '>' をキーとした単純な手法と比較して、訓練データで約 1/7、テストデータで約 1/4 に行単位の誤り率を減らすことができ、その有効性が確認された。

また、記号を 36 種に細分類したことにより、ブロック誤り率が約 1/6 になり、記号の細分類が非常に有効であることがわかった。ここで、記号を 4 種のみとした場合の決定木では、表 2の No.1 ～ 6, 8, 9, 12 の属性が用いられていた。これにより、ブロックのレイアウト的な情報は、記号の細分類の代用として十分でないともいえる。

さらに、準訓練データにおける主要な誤り原因を調べたところ、

- 引用記号なしの引用：5 ブロック（誤りの 35.7%）
- 空白文字のみの引用記号：3 ブロック（誤りの 21.4%）

であった。引用記号なしの引用は、ほとんどの場合、"--- ここから ---" のように、前後に引用の境界を示す行が挿入されているので、これらの行を検出することにより、ある程度判定できると考えられる。しかし、空白文字のみの引用記号は、単なるインデントである場合もあり、表層的な情報のみから判定するのは難しい。

3 内容種別の解析

内容種別は、1節で、通常文章、自動挿入文章、Signature、添付書類、図の領域の 5 種類からなると述べた。しかし、添付書類はフォーマットが定まっており、パターンマッチングで容易に特定できること、図は、表 1のコーパスにおいて 1 つしか存在しなかったことにより解析対象から除き、本稿では、通常文章、自動挿入文章、Signature の 3 領域の判定を行う手法を示す。

3.1 処理単位

内容種別は、その定義上、2節で設定された各引用構造の境界（図 1における点線）をまたいでは設定されない。また、空行で区切られることが比較的多いと考えられる。そこで、以下の手順により設定される "内容種別判定ブロック" を処理単位とすることにした。

1. ボディを引用構造の境界により分割する。
2. 引用領域では、引用記号を取り除く。
3. 2. の各領域を、空行により分割する。

3.2 内容種別判定

2節と同様に、C4.5 を用いて生成した決定木により内容種別の判定を行う。内容種別判定ブロックは、"通常文章"、"自動挿入文章"、"signature" に加え、これらが混合される場合も考慮して、"通常文章と signature の混合"、"自動挿入文章と通常文章の混合"、"自動挿入文章と signature の混合"、"通常文章と自動挿入文章と signature の混合" の 7 クラスに分類する。判定

表 5: 内容種別判定に用いる属性

No.	属性	値
1	ブロック行数	連続値
2	平均空白分割数／行	連続値
3	平均英字以外空白分割数／行	連続値
4	ブロック先頭文字種	12 種
5	ブロック末尾文字種	12 種
6	sig 境界定義行位置	有:3 種, 無
7	記号行位置	有:6 種, 無
8	ひらがな文字数	連続値
9	句読点数	連続値
10	文末表現数	連続値
11	From 行文字列有無	連続値
12	メールアドレス位置	有:2 種, 無
13	'in article' 行位置	連続値
14	ブロック末尾フレーズ	5 種
15	直前引用階層比較	=, <, >
16	直後引用階層比較	=, <, >
17	直前空行行数	連続値
18	直後空行行数	連続値
19	直前記号行ブロック位置	連続値
20	直後記号行ブロック位置	連続値
21	直前 sig 境界行ブロック位置	連続値
22	直後 sig 境界行ブロック位置	連続値
23	引用階層	連続値
24	ブロック逆順位置	連続値
25	同一元記事逆順位置	連続値
26	末尾からの行数	連続値

に用いる 26 属性を表 5に示す。以下、これらの属性を選択した着目点別にグループ分けして説明する。

ブロック内のレイアウト的な特徴を表す属性 属性 No.1 ～ 7 はブロック内部のレイアウト的な特徴を表す属性である。No.3 は、空白 or 改行で分割された英字以外の文字列数 / ブロック行数を表す。No.4, 5 で用いる文字種は、記号を句読点、シャープ、スペース、タブ、それ以外と分けた 12 種である。sig 境界定義行とは、signature の境界を表す行として推奨されている '-- ' と完全一致する行を表し、No.6 はそのブロック内位置情報として、ブロック先頭行、ブロック末尾行、中間行（先頭、末尾以外）、なしの 4 値を取る。記号行とは、英字、ひらがな、カタカナ、数字を全く含まない行を表し、No.7 は、記号行が 1 ブロック内で複数現れる場合も考慮して、そのブロック内の出現位置を、ブロック先頭行、ブロック末尾行、中間行、先頭行と末尾行、先頭行と中間行、末尾行と中間行、なしの 7 値とする。

文を表す属性 signature は単語列より構成されることが多い。このため、（単なる単語列ではなく）文を表すかどうかの指標となる表層情報を属性に加える（属性 No.8 ～ 10）。ひらがなは付属語である場合が多く、句読点は文末に置かれることが多いのでこれらのブロック内の総数を用いる（No.8, 9）。さらに、No.10 は、文末に現れることの多い、“ます”, “ません”, “ました”, “です”, “でした”, “でしょう”, “さい”, “以上” という表現を、ブロック内にいくつ含むかを表す。

投稿者情報を表す属性 signature は、投稿者に関する情報を記述しているという特徴がある。そこで、属性 No.11, 12 では、投稿者情報である可能性のある表現をもつかどうかを表す。No.11 は、ヘッダの From フィールドより、セパレータである '()<>' を取り除き、スペース単位に分割した文字列[4]が、ブロック内に含まれる数を表す。No.12 は、ブロック内にメールアドレスと考えられる文字列を含むか、含む場合には、No.10 の表現の前か後ろか（文中にあるかないか）の 3 値をとる。

自動挿入文章の特徴的表現を表す属性 属性 No.13, 14 は、自動挿入文章に含まれることの多い表現を含むかどうかを表す。No.13 は、'in article' という文字列がブロック内の何行目に現れるかを表す（なければ 0）。No.14 は、ブロック末尾から記号を除いた文字列を、自動挿入文章の末尾に現れることの多い次の 4 表現、“書く”（“書きました”, “曰く” 等日本語表現）、“write”（“wrote”, “says” 等英語表現）、“さん”、“article” と “その他” に分類する。

ブロック前後環境を表す属性 属性 No.15 ～ 22 は、ブロックの前後環境を表す属性である。No.15, 16 は、当該ブロックが直前 / 直後のブロックと比較して、引用の階層が同じレベ

[4] “Yamada Taro <yamada@test.mail>” では、“Yamada”, “Taro”, “yamada@test.mail”

ル、より深い、より浅いのいずれかであるかを表す。No.19, 20は、記号行を含むブロック（最も近いもの）が、当該ブロックの前方/後方何ブロック目にあるかを表す（なければ0）。ただし、検索対象ブロックは、異なる引用階層となるブロックが現れるまでである。No.21, 22は同様に、sig境界行が含まれているブロックが何ブロック目にあるかを表す。

ブロックの位置情報を表す属性 属性No.23～26は、ブロックの位置情報を表す属性である。No.23は、引用の深さを表す。No.24は、全ブロックの末尾からの位置を表す。No.25は、同一の元記事のブロックを集め、そのブロック群末尾からの位置を表す。No.26は、ブロック先頭行が、ボディ末尾から何行目にあるかを表す。

3.3 混合ブロックの分割

ブロックが、2つ以上の内容種別の混合と判定された場合（これを混合ブロックとよぶ）には、それらを1つの内容種別単位に分割する必要がある。本稿では、混合ブロックの学習データ数が少なかったことから、"通常文章とsignatureの混合"、"自動挿入文章と通常文章の混合"について、ヒューリスティックによる分割を検討した。[5]

3.3.1 通常文章とsignatureの分割

signatureは通常同一引用階層の末尾につけられる。そこで、"通常文章とsignatureの混合"は、通常文章、signatureの順に並んでいるとして、分割処理を行う。

1. signature境界定義行('-- ')が存在した場合には、その行以下をsignature、それより上の行を通常文章とする。

2. 記号行が存在した場合には、（複数存在する場合には最も上の）その行以下をsignature、それより上の行を通常文章とする。

3. 行末が句読点の行が存在した場合には、（複数ある場合には最も下の）その行より下をsignature、その行以上を通常文章とする。

4. 句読点を含む行が存在した場合には、（複数ある場合には最も下の）その行より下をsignature、その行以上を通常文章とする。

5. 1行目を通常文章、それ以外をsignatureとする。

3.3.2 自動挿入文章と通常文章の分割

自動挿入文章は、引用領域の直前に現れることが最も多い。ここで、引用領域の境界はブロックの末尾となるため、自動挿入文章の後ろに通常文章がくることはほとんどないと考えられる。そこで、"自動挿入文章と通常文章の混合"は、通常文章、自動挿入文章の順に並んでいるとして、分割処理を行う。

1. ブロック行数が1行、または、'in article'行が1行目の場合は、ブロック全体を自動挿入文章とする。

2. 'in article'行が存在する場合には、それより上の行を通常文章、それ以下の行を自動挿入文章とする。

3. ブロック末尾行を自動挿入文章、それ以外を通常文章とする。

3.4 実験

提案した内容種別判別法の評価を行うため、表3の訓練データを用いて決定木を生成した。

3.4.1 作成された決定木の特徴

表3の訓練データ（8595ブロック）により作成された決定木の特徴は以下の2点であった。

- 訓練データ中に、"自動挿入文章とsignatureの混合"は1ブロックのみ、"通常文章と自動挿入文章とsignatureの混合"は存在しなかったため、この決定木でこれらのクラスに判定されるブロックは存在しなかった。これは、

[5] 3.4.1節に示すように、実験で生成された決定木では、他の混合ブロックには分類されないので省略する。

表 6: 内容種別判定精度

データ	訓練	テスト	訓練					
除いた属性	-	-	No.1-7	No.8-10	No.11-12	No.13-14	No.15-22	No.23-26
全ブロック数	8595	967	8595					
全行数	23209	2532	23209					
誤りブロック数	81	12	167	104	108	109	106	120
ブロック誤り率	0.94%	1.24%	1.94%	1.21%	1.26%	1.27%	1.23%	1.39%
誤り行数	149	33	363	178	252	186	184	220
行誤り率	0.64%	1.30%	1.56%	0.77%	1.09%	0.80%	0.79%	0.95%

自動挿入文章は引用の直前に現れる、signature は末尾に現れることを間接的に示しているといえる。

- 決定木のノードとして利用されなかった属性は、No.4, 5, 18, 23 であった。

3.4.2 判定精度

表 6に、提案手法の訓練データとテストデータに対する判定精度[6]、および、各属性の有効性を検証するために、3.2節で述べた属性グループから、1グループずつ除いた場合の訓練データの判定精度を示す。"通常文章と signature の混合"、"自動挿入文章と通常文章の混合" と判定されたブロックには、3.3.2節の分割ルールを適用する。ここで分割位置を誤った場合には、行誤り数が増加する。

3.4.3 考察

表 6より、訓練データ、テストデータ共に低い誤り率であり、本手法の有効性が確認できた。

属性として除くことにより、誤りブロック数が大きく増加した属性グループは、"ブロック内のレイアウト的な特徴を表す属性" (属性 No.1 ～ 7) と "ブロックの位置情報を表す属性" (属性 No.23 ～ 26) であり、内容種別の判定ではこれらの属性が特に有効であることがわかった。

4 おわりに

本稿では、決定木を用いて、改行等により区切られたブロックという単位で引用構造および内容種別を解析する方法について述べた。そして引用構造の解析には、引用記号候補列の特徴を表す情報が重要であり、特に記号を細分類した文字種が有効であること、内容種別の解析では、ブロック内のレイアウト的な情報とブロックの位置情報が重要であることを示した。本手法で解析された構造は、[2] で示した XML タグのサブセット (処理区分領域用タグおよび一部の元テキスト再現用タグ) が付与されて出力される。今後は、"処理単位" 構造の自動解析について検討を進めていく。

[6] 全行数は、空行 (引用記号を除いた時に空行となったものも含む) を含めなかったため、表 4と値が異なる。

参考文献

[1] 浅野久子, 加藤恒昭, 高木伸一郎. Signature の局所的パターンマッチによる電子メールからの送信元住所録情報の抽出とそれを用いた住所録管理システム. 情報処理学会論文誌, Vol. 39, No. 7, pp. 2196–2206, 1998.

[2] 浅野久子, 永田昌明. ネットニュース用 XML タグセットの検討とその構造解析への応用. 言語処理学会第 4 回年次大会予稿集, pp. 460–463, 1998.

[3] 長谷川隆明, 高木伸一郎. 電子メールコミュニケーションにおけるスケジュール情報抽出. 情報処理学会研究報告, NL123-10, pp. 73–80, 1998.

[4] J. Ross Quinlan. *C4.5 Programs for machine learning*. Morgan Kaufmann Publishers, 1993.

[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, pp. 2371–2379, 1995.