文字列の出現頻度情報を用いた分かち書き単位の自動取得

岡田 正平　　山本 和英
長岡技術科学大学 電気系
{okada, yamamoto}@jnlp.org

1　序論

自然言語によって書かれたテキストを処理する際，対象のテキストはその文ごとに形態素や文節といった何らかの単位によって分割される．その際，特に日本語などの分かち書きされていない言語では，その分割基準の曖昧性が問題になる．この問題を解決するため現存の多くの形態素解析器は，人手によって作られた辞書と教師データから学習を行っている．ところがその教師データは新聞記事をもとに作られていることが多いため，blog 記事などに見られる口語体で書かれたテキストに対して充分な精度を発揮することができない．また，新しい語や表現は日々生まれ続けており，これらに対応できるような辞書や教師データを作成するにはコストが掛かる．そもそも辞書や教師データの作成にあたっては複数の基準や品詞体系が存在しており，何をもって正解とするのかが曖昧な部分もある．

これらの問題を受け，我々は事前に辞書や教師データを必要としない分割を行うことを目的とした．我々の手法では辞書を必要としないので，既知語・未知語に拘らず分割可能になる．また，教師データの存在しない言語のテキストも分割することができる．

本研究ではその前段階として，テキスト中の文字列の出現頻度のみを用いた分割単位の自動取得を試みた．その手法として分割単位を文字列の定型表現とみなし，定型表現の自動抽出手法を文字列に対して適用した．

新聞記事に対して実験を行った結果，通常の単語分割では細分化されてしまうような複合名詞などを分割単位として取得した．

2　関連研究

岡野原ら [1] や持橋ら [2] は、日本語における分割単位（単語）の問題について言及している．

岡野原ら [1] は，文書分類タスクにおいて有効に働く文字列は単語とは異る場合があることを述べ，テキスト中の全ての部分文字列を素性とした文書分類モデルを提案した．持橋ら [2] は既存の形態素解析器が口語体の文章や，教師データの存在しない古文などに対して適切な分割を行えないことを挙げ，教師データを必要としない，ベイズ学習に基づく形態素解析手法を提案した．

同様に分かち書きされていない言語である中国語において，Zhao [3] は単語分割の基準がいくつも存在するという問題を挙げ，単語に代って文字間の係り受け関係を用いてテキストの構造を表現した．

本研究では文の分割を行う前段階として分割単位のリストを自動取得する．このリストは単語の辞書のようなものとして扱うことができるので，事前に辞書を必要としなくなる．手法としては分割単位を文字列の定型表現とみなして自動抽出を行う．事前に辞書や教師データを必要とせず定型表現を自動抽出する方法として，北ら [4] は仕事量基準という概念を導入し，テキスト中の各単語列にスコアを付ける方法を提案した．本研究ではこの方法を文字列に対して適用することで，文字列の定型表現として分割単位の自動取得を試みる．

3　分割単位の自動取得

北ら [4] は仕事量基準の削減量という観点からテキスト中の単語列に対してスコア付けを行い，その結果から更にスコアの再計算を行った．また，計算量を削減するために再計算時には条件付けを行った．本研究ではこれに基づき，テキスト中の文字列に対して以下の手順でスコア付けを行う[1]．

まず，テキスト中に出現する全ての文字列 s に対して次式にてスコアを付ける．

$$\mathrm{score}(s) = \mathrm{len}(s) \times \mathrm{freq}(s) \qquad (1)$$

[1]本研究の目的は北らが導入した「仕事量」の削減ではない．しかし，辞書や教師データを必要としない定型表現の自動抽出方法という点が本研究の目的に合致したためこの手法を採用した．

文字列 s	freq(s)	score(s)		文字列 s	freq(s)	score(s)
ab	200	400	再計算 ⇨	ab	~~200~~ 110	~~400~~ 220
abcd	80	320		abcd	80	320
abd	90	300		abd	90	300

(a) 再計算は隣り合った文字列同士に限る.ab と abcd では長さが 2 異なるため，この分の出現頻度を減じない.

文字列 s	freq(s)	score(s)		文字列 s	freq(s)	score(s)
ab	200	400	再計算 ⇨	ab	~~200~~ 100	~~400~~ 200
abc	100	300		abc	100	300
abd	40	120		abd	40	120

(b) 再計算は一度だけに限る．ab の出現頻度は既に abc の分を引いているので abd の分は減じない

図 1: 再計算時の条件

ここで len(s) は s の長さ（文字数），freq(s) は s のテキスト中の出現回数を表す[2]．このスコアが高い方から上位の文字列を分割単位として採用していく．

しかしテキスト中には「インターネット」という文字列とほぼ同等の回数「インターネッ」や「ンターネット」という文字列も出現する．単にこのスコアに基づくと，特に長い文字列について上記のような部分文字列を採用してしまう可能性が大きい．そこで一度スコア計算を行った後に部分文字列の出現回数を補正し，スコアの再計算を行う．

ある文字列 s_1 がある文字列 s_2 の部分文字列である場合，s_1 の出現回数 freq(s_1) には s_2 の出現回数 freq(s_2) が含まれている．したがって s_2 が分割単位として採用されている場合，純粋な s_1 の出現回数 freq$'$(s_1) は

$$\mathrm{freq}'(s_1) = \mathrm{freq}(s_1) - \mathrm{freq}(s_2) \quad (2)$$

であると考えられる．よって s_1 のスコアは

$$\begin{aligned} \mathrm{score}(s_1) &= \mathrm{len}(s_1) \times \mathrm{freq}'(s_1) \\ &= \mathrm{len}(s_1) \times \{\mathrm{freq}(s_1) - \mathrm{freq}(s_2)\} \quad (3) \end{aligned}$$

として再計算する．

この再計算をすべての部分文字列に対して行っていくのだが，厳密にすべての文字列に対して行うと計算量が膨大になってしまう．そこで，以下に示す条件を付与して計算量を削減する．

再計算は隣り合った文字列同士に限る

隣り合った文字列とは，長さが 1 しか異ならない 2 つの文字列のことである．つまり，文字列 abcd が分割単位に採用されても文字列 ab に対して再計算は行わないこととする．(図 1(a))

再計算は一度だけに限る

再計算の際にはスコアの高い文字列から順に見ていき，その部分文字列の出現回数を減算していく．このときの減算はひとつの文字列に対して一度だけに限る．つまりスコア順に文字列 abc，abd が採用されていても，文字列 ab の出現頻度は abc の分しか減算しないこととする．(図 1(b))

[2]通常の定型表現は複数の単語からなるため，それを抽出する場合は単語列の長さから 1 を引いて計算を行う．それに対し分割単位は 1 文字でもあり得るため，本手法では文字列の長さのまま計算を行う．

4 分割単位の取得実験

日本経済新聞記事 1 年分 (1)（2000 年版，194,337 記事）を用いて分割単位を自動取得する実験を行った．上記手法に基づいて長さ 20 までの文字列にスコア付けを行い，分割単位として採用する文字列を 50 万，100 万，200 万と変化させながら結果を比較した．計算の際には各記事間の内容は独立であると考え，記事間をまたぐ文字列は計算の対象外とした．

実験の結果，分割単位として採用した数を変化させてもスコア順に並べた際の上位の文字列にはほとんど差が無かった．100 万の文字列を採用した場合と 200 万の文字列を採用した場合を比較すると上位 7382 位までの文字列が一致し，50 万の文字列を採用した場合と 100 万の文字列を採用した場合を比較すると上位 3213 位までの文字列が一致した（ゆえに 50 万の場合と 200 万の場合でも同様に上位 3213 位まで一致した）．取得された文字列のうち，文字数別（1 文字から 4 文字まで）の上位 10 位までの文字列とそのスコア，全体での順位を表 1 に示す．得られた分割単位には長い文字列ほど上位に入りにくくなる傾向が見られた．上位 15 位まではすべて 1 文字であり，上位 100 位までもすべて 5 文字以内の文字列だった．取得された比較的長い文字列の例を，上位 200 万の文字列を採用した場合の順位・スコアと共に表 2 に示す．

5 得られた分割単位による文の分割

得られた分割単位を用いて入力文を分割する実験を行った．単語のリストのみからテキストの分割を行う方法には形態素数最小法がある．本実験ではこれに倣い，得られた処理単位での分割数が最も少なくなるような方法で分割を行った．また，分割単位として取得

表 1: 分割単位として取得された文字列のうち，文字数別のスコア上位 10 位までの文字列

(a) 1 文字

順位	s	score(s)
1	の	2460846
2	、	1698782
3	に	1411488
4	を	1386141
5	は	1145981
6	。	1100211
7	が	1019633
8	る	967110
9	と	934911
10	で	866192

(b) 2 文字

順位	s	score(s)
16	する	611690
19	る。	567434
22	た。	513134
25	した	436938
26	など	436562
30	から	390454
32	して	354524
43	った	305054
62	ない	227848
64	で、	225898

(c) 3 文字

順位	s	score(s)
27	した。	425859
37	ている	328968
46	する。	296121
51	った。	271755
80	として	201654
112	という	142488
123	などの	134955
130	ること	126450
132	してい	125268
156	などを	111921

(d) 4 文字

順位	s	score(s)
40	ている。	317820
91	している	181120
150	について	115384
168	という。	104788
175	サービス	101324
183	システム	98912
200	っている	92952
234	だった。	81900
242	すること	79508
278	センター	69568

表 2: 取得された比較的長い語とその順位（上位 200 万の文字列を採用した場合）

順位	s	score(s)	順位	s	score(s)
188	インターネット	96803	1459	となっている。	19943
658	平方メートル	36600	1908	なければならない	15792
762	すると発表した。	32960	2647	朝鮮民主主義人民共和国（北朝鮮）	11760
768	情報技術（ＩＴ）	32832	3232	コンビニエンスストア	9920
799	コンピューター	32270	3789	主要国首脳会議（沖縄サミット）	8580
1391	二〇〇〇年三月期	20664	4875	米店頭株式市場（ナスダック）総合指数	6930

※ () 内の文字も含めてひとつの分割単位として取得されている．

されていない文字列が入力文中に出現した場合には，その部分をひとかたまりとした．

入力文と，各採用数ごとの出力の例を，形態素解析器 MeCab[(2)] による単語分かち書き結果と共に以下に示す[3]．ただし可読性を上げるため，処理単位間の区切りを/で表現している．

入力テキスト

年の瀬も迫った三十日、九州各地の商店街やスーパーはお節料理の材料やしめ縄など正月用品を買い求める客でにぎわった。

上位 50 万の文字列を採用した場合

年/の瀬/も/迫っ/た三十/日、九/州各地/の商店街/やスーパー/はお/節/料理の/材料や/しめ/縄/など/正月/用/品を買/い求める/客でにぎわ/った。

[3]この例には，分割単位として取得されていない文字列は含まれていなかった．

上位 100 万の文字列を採用した場合

年/の瀬/も/迫った/三十日、/九州各地/の商店街/やスーパー/はお/節/料理の/材料や/しめ/縄/など/正月/用品/を買い求め/る客/でにぎわった。

上位 200 万の文字列を採用した場合

年の瀬/も/迫った/三十日/、九州各地/の商店街や/スーパーは/お節/料理の/材料や/しめ/縄など/正月/用品/を買い求め/る客/でにぎわった。

MeCab による単語分かち書き

年の瀬/も/迫っ/た/三/十/日/、/九州/各地/の/商店/街/や/スーパー/は/お節/料理/の/材料/や/しめ縄/など/正月/用品/を/買い求める/客/で/にぎわっ/た/。

MeCab による分かち書き結果では「三十日」という文字列を「三/十/日」と細分化しているのに対し，上位 200 万の文字列を採用した場合には「三十日」と

ひとつのかたまりになっている．また「年の瀬」という文字列に対しては，採用数を増やすことによってひとつのかたまりに分割するようになった．

6　考察

実験の結果から，通常の単語分割では細分化されてしまう複合名詞などをひとつの分割単位として扱えるようになると考えられる．しかし，今回の取得方法では長い文字列ほど上位に入りにくい結果となった．分割単位として採用する文字列の数を増やせば採用される長い文字列も増えるが，その分意味のない文字列も多く取得してしまう．そのため，長い文字列のスコアが高くなるように手法を検討する必要がある．長い文字列ほど上位に入りにくくなっている理由には，計算量削減のための再計算の条件付けが挙げられる．条件付けによって部分文字列になっている短い文字列のスコアが引かれにくくなっているためである．また，スコアの計算式が出現頻度と長さの単純な積であることも挙げられる．長い文字列がより多く採用されるには，文字列の長さと出現頻度の関係を計算式に反映させるべきである．

テキストの分割実験では，既存の分割単位における文節にあたる文字列で分割されている部分が見られる．今回の用いた分割数を最小にする分割方法では，より小さな部分文字列（例えば単語にあたる文字列）が採用されていたとしても，高い確率で長い方の文字列に分割されるためである．

本研究においては，結果が既存の単語分割に一致することが必ずしも正解であるとは言えない．得られた分割単位の有用性を評価するためには，実際の言語処理タスクへの応用や，パープレキシティの計算といった定量的な評価を行う必要がある．

7　結論

本研究では辞書や教師データを必要としない方法を用いて，分かち書き単位の自動取得を試みた．本手法では辞書や教師データを必要としないことから，未知語・既知語関係無く処理を行うことができるようになる．また，教師データの存在しない言語に対してもテキストの分割を行うことができる．具体的にはテキスト中に出現する全ての文字列に対して，その長さと出現頻度の積によるスコア付けを行い，スコア上位のものから分割単位として採用していった．

新聞記事に対して実験を行った結果，得られた上位のものには短い文字列が多く見られた．しかし，より下位には通常では細分化されてしまう複合名詞のような長い文字列も得られていることも確認できた．

8　今後の展望

通常では細分化されてしまう，複合名詞などの長い文字列をより多く取得できように手法を検討する．また，得られた分割単位の有用性を評価するために，実際のタスクへの応用や定量的な評価を行う．

今回の実験は新聞記事に対して実験を行ったが，今後は口語表現やWeb上のテキストに対して実験を行った場合，どのように影響するかを観察する．

使用した言語資源およびツール

(1) 日本経済新聞全記事データベース CD-ROM 版，2000 年版

(2) 形態素解析器 MeCab，Ver.0.993，工藤拓，
`http://mecab.sourceforge.net/`

参考文献

[1] 岡野原大輔, 辻井潤一. 全ての部分文字列を考慮した文書分類 (分類). 情報処理学会研究報告. 自然言語処理研究会報告, Vol. 2008, No. 90, pp. 59–64, sep 2008.

[2] 持橋大地, 山田武士, 上田修功. ベイズ階層言語モデルによる教師なし形態素解析 (言語モデル・ウェブ解析). 情報処理学会研究報告. 自然言語処理研究会報告, Vol. 2009, No. 36, p. 49, mar 2009.

[3] Hai Zhao. Character-Level Dependencies in Chinese: Usefulness and Learning. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pp. 879–887, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[4] 北研二, 小倉健太郎, 森元逞, 矢野米雄. 仕事量基準を用いたコーパスからの定型表現の自動抽出. 情報処理学会論文誌, Vol. 34, No. 9, pp. 1937–1943, sep 1993.