A-020

形態素解析と文法圧縮を利用した 日本語テキストに対する圧縮手法の一検討

A Study on Japanese Text Compression Using Morphological Analysis and Grammar Compression

中村 公美† 河野 和宏‡ 馬場口 登†
Kumi Nakamura Kazuhiro Kono Noboru Babaguchi

1. はじめに

テキストデータを圧縮してデータ量を削減するだけでなく，圧縮したままの状態であっても文字列検索を可能とする圧縮手法の一つとして，「単語に基づくテキスト圧縮法(Word-based text compression)」[1]が知られている．ただし，単語に基づいて符号化するため，英語テキストのような，単語間がスペースで区切られたテキストであれば直接適用できるものの，日本語テキストのような単語間に区切りがないテキストの場合，圧縮法を適用する前に何らかの手法で区切りを入れる必要がある．

Yoshida ら[2]は形態素解析を用いて日本語テキスト内の各文章に区切りを入れた後，各単語を符号化する単語に基づくテキスト圧縮法を提案している．形態素解析を文書の区切りに利用する場合，符号化される単語は形態素単位で区切られた単語であることから，圧縮したままの単語検索に適している．一方，形態素には長い単語があるように，圧縮を踏まえた区切りとは必ずしもなっていないため，圧縮テキストのファイルサイズに改善の余地があるといえる．

別の手法として，正木ら[3]は Re-Pair[4]と呼ばれる文法圧縮の手法を応用して，形態素解析や構文解析を用いずに日本語テキスト内の各文章に区切りを入れた後 End-Tagged Dense 符号(ETDC)[5]で符号化するテキスト圧縮法を提案している．文法圧縮を用いる場合，単語の出現頻度を見て区切りを入れることになるため，符号化後の圧縮テキストのファイルサイズは形態素解析を用いる手法よりも小さくなると考えられる．一方で，自然言語的に区切ることはないため，文節が正確でなく，文字列検索が難しくなる．

そこで本稿では，まず形態素解析により区切りを入れた後，そのまま符号化するのではなく Re-Pair を用いて文法変換し，得られた文法を ETDC により符号化する手法を提案する．形態素解析を用いて区切りを入れた単語を利用することで単語の検索処理が容易となること，Re-Pair で文法変換することにより符号化後の圧縮テキストの圧縮率の向上が期待される．

2. 関係する用語とアルゴリズムの概説

2.1 文法圧縮

文法圧縮とは，与えられた文字列Sを一意に表す文法に変換することでSを圧縮する方法である[6].この時構築される文法の多くは文脈自由文法 (context-free grammar(CFG))である．CFG は$G=(V,\Sigma,P,s)$の形で表される．V，Σ，Pはそれぞれ，非終端記号，終端記号，生成規則$\alpha \to \beta$の集合を，sは開始記号を示す．$\alpha \in V, \beta \in (\Sigma \cup V)^*$である．

†大阪大学大学院工学研究科 Graduate School of Engineering, Osaka University
‡関西大学社会安全学部 Faculty of Societal Safety Sciences, Kansai University

2.2 Re-Pair

Re-Pair [4]は与えられた文字列Sに対し，①入力された文字列内で最も出現する回数の多い隣接する記号ペアを発見する，②その記号のペアを新たな非終端記号に置換する，③その置き換え規則をPに追加する，ことにより新たな文法に変換する．その後，置き換え後の文字列を新たな入力文字列とし，隣接する記号ペアの出現頻度がすべて 1 になるまで繰り返す．Re-Pair の概要を図 1 に示す．

2.3 End-Tagged Dense 符号(ETDC)

単語の出現回数が多い順番に短い符号を割り当てる符号化を Dense 符号といい，ETDC [5]では n ビット単位可変長符号を割り当てることで符号化する．$1\times n$ から $m\times n$ ビットの範囲で符号を割り当てる場合，符号内の最後の n ビット列の最初のビットを 1 とし，それ以外のビット列の最初のビットを 0 とするような符号を生成し，割り当てることで，符号化後のテキストにおける符号の切れ目をわかりやすくすることができる．図 2 に 2 ビット単位で ETDC を用いて符号化する例を示す．

3. 提案手法

与えられた日本語テキストSに対し，提案手法では，まず形態素解析によりN個の文字列に区切る．このとき，Sの区切り後の文字列群を$T=\{t_1,t_2,\ldots,t_N\}$とすると，TはM種類の単語$W=\{w_1,w_2,\ldots,w_M\}$の組み合わせで示されることになる．その後，W に含まれる単語を 1 単位として区切られたTに対し，各w_i を終端記号として Re-Pair を行い，非終端記号の集合Σ及び生成規則の集合Pを生成し，文法Gを構築する．最後に，Gに対して ETDC を用いて符号化して，圧縮テキストを得る．「青巻き紙赤巻き紙黄巻き紙黄巻き紙赤巻き紙」でGを構築した例を図 3 に示す．文法を視覚的に表すため，生成規則における右辺を親，左辺を子とする形で構文木として示している．

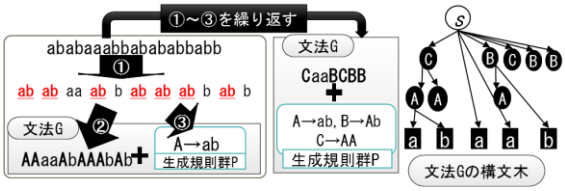

図 1 Re-Pair の概要図

頻度の順位	記号	符号
1	A	10
2	B	11
3	C	00 10
4	D	00 11
5	E	01 10

00 10 10 11 01 10 10…
[00 10] [10] [11] [01 10] [10]…
CABEA…
復号プロセスの例

図 2 ETDC による符号割り当てと復号プロセスの例

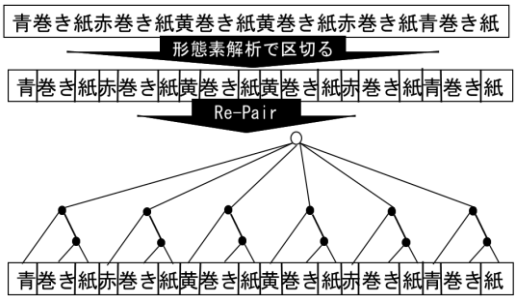

図 3　提案手法での文法の構築例

ただし，隣接する記号によって成り立つペアのS内における出現頻度が 1 になるまで置き換えを繰り返していけば，計算コストが嵩むうえに，記号が増える，すなわちGを符号化し,保存する際のサイズが大きくなる．そこで，この再帰回数に対して一定の基準を設け，その基準が満たされたときに再帰処理を打ち切る．

j回目までに追加されるj個の非終端記号の集合Σ_jとM種類の終端記号Wをまとめて$\Sigma_j \cup W = \{m_1, m_2 \ldots, m_{M+j}\}$とする．$j$回目の再帰における，$j$-1 回目の圧縮テキストのサイズからの増減分$\Delta_j$を以下の式に従って概算する．

$$\begin{cases} \Delta_j = \sum_{i=1}^{M+j} \left|\varepsilon\left(o_i^j\right)\right| - \sum_{i=1}^{M+j-1} \left|\varepsilon\left(o_i^{j-1}\right)\right| + \left|\varepsilon(o_x^j)\right| + \left|\varepsilon(o_y^j)\right| \\ o_x^j = o_x^{j-1} - a, o_y^j = o_y^{j-1} - a, o_{m+j}^j = a \ \ (x \neq y) \\ o_x^j = o_x^{j-1} - 2a, o_{m+j}^j = a \ \ (x = y) \end{cases}$$

ただし，j回目に置き換えが行われる記号のペアを$m_x m_y$とし，その出現頻度をaとしている．また，$\varepsilon\left(o_i^j\right)$は$j$回目の繰り返しで再帰処理を終了したと仮定したときにm_iに ETDC を用いて割り当てられる符号を示し，$\left|\varepsilon(o_i^j)\right|$はその符号の長さを示す．$\Delta_i$について，$j-2 \leq i \leq j$のとき$\Delta_i > 0$ ならば再帰処理を打ち切る．

4. 実験方法

提案手法を評価するため，実験を行った．利用したデータセットは以下の通りである．

- Wikipedia より 10 記事…ウィキペディア，ZIP_(ファイルフォーマット)，大阪大学，夏目漱石，コンピュータ，プログラミング言語，万年筆，データ圧縮，地方病_(日本住血吸虫症)，三毛別羆事件
- 青空文庫より夏目漱石による著作物 10 作品…こころ，それから，余と万年筆，吾輩は猫である，坊っちゃん，文士の生活，自転車日記，草枕，虚子君へ，趣味の遺伝

また，比較手法として，一文字ずつ区切る手法，[2]に基づき形態素解析(kagome[7])で区切る手法，[3]に基づき Re-Pair を応用して区切る手法を用いた．符号化法は全て ETDC とした．

5. 実験結果と考察

提案手法および比較手法の圧縮率の平均，最大，最小を表 1 に示す．なお，圧縮率は「圧縮後のファイルサイズ/圧縮前のファイルサイズ」で定義される．CFG を ETDC で符号化した辞書ファイルと，圧縮前のテキストを辞書に従って符号化したファイルの 2 ファイルの合計を圧縮前のファイルサイズで割ったものである．圧縮率が低いほうが性能がよいといえる．

表 1　提案手法の圧縮率が最小・最大であるときの圧縮率

	文書名	一文字区切り	Re-Pair での区切り	形態素解析での区切り	提案手法
平均値		56.4%	51.4%	59.4%	60.2%
最小	こころ	40.3%	35.9%	44.3%	37.1%
最大	余と万年筆	62.8%	67.3%	77.5%	83.7%

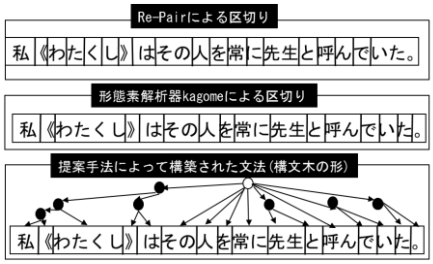

図 4　各手法による文の一部の区切り及び生成される文法例

こころは約 559 キロバイトであるのに対し，余と万年筆は約 11 キロバイトのテキストデータである．また，提案手法の中で 2 番目に最小となった圧縮率のテキストデータは 1MB を越える「吾輩は猫である」であり，39.3%であった．この時 Re-Pair のみによる区切りの手法での圧縮率も 39.2%とほぼ同程度であった．以上より，十分な大きさのある文書であれば，提案手法の圧縮率は優れていると考えられる．

次に，データ圧縮の結果，どのような文法が構築されているかの一例を図 4 に示す．図 4 は「こころ」の最初の一文「私《わたくし》はその人を常に先生と呼んでいた。」であり，提案手法の図中の黒丸は非終端記号を示す．図 4 より，Re-Pair での区切りでは，自然言語的に不自然な区切りがなされているのが分かる．また，形態素解析のみを利用した場合と比較して，提案手法では何度も出てきている単語が新たに一つの単語として構築されており，より検索に向いた構文木が作成されていることがわかる．

6. 結論

本稿では，日本語テキストに対して形態素解析を用いて区切りを挿入した後，区切りの各単語を終端記号として Re-Pair を適用して文法変換し，最後に ETDC を用いて符号化して圧縮テキストを生成する手法を提案した．今後の課題として，提案手法が文字列検索に有効であることを示すために，様々な文字列検索手法を実装して検証する必要がある．なお，本研究の一部は科学研究費補助金による．

参考文献

[1] Moffmat, A., “Word-based text compression”, Software: Practice and Experience, Vol. 19, Issue 2, pp. 185-198(1989).
[2] Yoshida, S., Morihara, T., Yahagi, H., Itani, N., “Application of a Word-Based Text Compression Method to Japanese and Chinese Texts”, IEICE Trans. On Fundamentals of Electronics, Communications and Computer Sciences, Vol. E85-A, No. 12, pp. 2933-2938(2002).
[3] 正木ら，“日本語テキストに対する検索指向符号化のための文法圧縮分割”,情報科学技術フォーラム(FIT)講演論文集, Vol. 13, No. 1, pp. 67-68,(2014).
[4] Moffmat, A., “Off-line dictionary-based compression”, Proceedings of the IEEE, Vol. 88, Issue 11, pp. 1722-1732(2000).
[5] Brisaboa, N., Iglesias, E., Navarro, G., Paramá, J., ”An Efficient compression code for text database”, Proceedings of the 25th European conference on IR research, pp. 468-481(2003).
[6] 田部井靖生, “文法圧縮最前線”, 情報処理, Vol. 5, No. 2, pp. 172-178(2016).
[7] Ikawaha, “Kagome Japanese Morphological Analyzer”, https://github.com/ikawaha/kagome