# 単語を単位とした文書間差分抽出方式およびその高速化手法

4S－8

青山 ゆき　東野 純一
(株) 日立製作所 システム開発研究所

## 1. はじめに

ワープロ等の上で、旧文書より新文書を編集する際に、新旧文書の差分文字列を機械的に把握することは、編集作業の効率化につながる。しかし、一般にワープロソフトの文書比較機能は、文書の段落、行等のある程度の文字列のまとまりで比較しており、頻繁に使用される単語を置換した場合などでは差分箇所を把握し難い。そこで、単語を単位とした差分を抽出することが適切であるが、日本語文書の単語分割処理の計算量は大きく、単語単位の差分を迅速に抽出することは困難であった。

本稿では旧文書のみ単語分割し、分割されていない新文書との差分を抽出する〔単語－文字列〕間差分抽出方式を提案する。これにより、差分抽出時に単語分割することなく、迅速に単語単位の差分を抽出できる。また、従来の2種類の差分抽出手法を〔単語－文字列〕間に拡張した方式を組み合わせて、高速に差分抽出する手法ついて述べる。

## 2. 単語単位の差分抽出方式

「単語」を最小単位として差分を抽出する方式として、まず次の2つが考えられる。

(1) 前処理方式：比較対象の両文書を単語分割した後、単語を単位に文書間の比較を行う。

(2) 後処理方式：文字単位で両文書間の比較を行った後、差分として抽出された文字を単語単位の差分に修正する。

前処理方式は、差分抽出の前に両文書全体を単語分割することが必要であるが、日本語文書の単語分割処理の計算量は大きく、対話的にユーザーに差分情報を与えたい場合には適さない。また、後処理方式を用いると、文字単位の差分抽出では、

新) 前述の日程では、他の
旧) 前日までの

のような、細切れの結果が抽出されてしまうことがある(下線部が差分)。このような差分が文書全体に広がった場合、単語単位の差分に修正することは困難である。

そこで、前処理方式における単語分割の負荷の大きさの問題を解決するために、〔単語－文字列〕間での差分抽出方式を採用した。すなわち図1に示すように、以下の手順に従って旧文書と新文書の差分を抽出する。

手順1) 旧文書を予め単語分割し、既分割旧文書として保存しておく。

手順2) 旧文書から新文書を編集する。

手順3) 差分抽出部が呼び出された時点で既分割旧文書を読み込み、分割された旧文書と通常の分割されていない新文書間で、〔単語－文字列〕間差分抽出を行う。

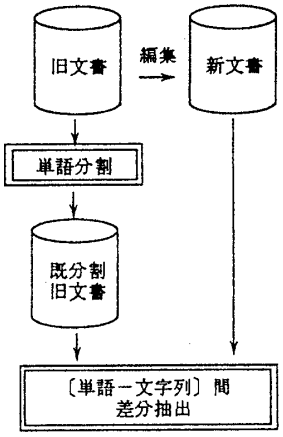


図1 〔単語－文字列〕間差分抽出の流れ

この方式を取ることにより、文書編集中に差分抽出が呼び出された時点で、単語分割することなく、迅速に単語単位の差分を抽出することが可能となる。

## 3. 2段階差分抽出方式

差分抽出方式として広く使われているものに、Heckel法およびLCS法がある[1]。これらの手法を簡単に説明し、さらに組み合せて高速化する手法について述べる。

### 3.1 Heckel法

Heckel法とは2つの文書間の唯一である要素をkeyとして対応付けを行う手法である。

＜Heckel法のアルゴリズム＞

[Step1] 新旧の文字列に対し、唯一の要素同士を対応づける。

[Step2] 唯一要素の前後を辿り、同一要素かつ未対応であれば対応づける。

例えば「ABCDE」と「ABCADF」を比較した場合、[Step1]で"B","C","D"を唯一の要素として対応付け、[Step2]で"B"の前の"A"を対応付ける。よって「ABCDE」と「ABCADF」(下線部が差分)となる。

### 3.2 LCS法

各種のファイル間比較プログラムで頻繁に使われる手法に最長共通部分列(LCS: Longest Common Subsequence)を求めるLCS法がある。

＜LCS法のアルゴリズム＞

[Step1] 新旧の文字列に対し、一致する要素(唯一とは限らない)の組合せを見つける。

[Step2] 一致する要素をつなぎ、共通部分列をつくる。

---

A Improved Method of Extracting Different Words Between Docquments
Yuki Aoyama and Junichi Higashino
System Development Laboratory, Hitachi, Ltd.

[Step3] [Step2]のうち最長の共通部分列(LCS)を求める。

「ABCDE」と「ABCADF」比較した場合は、[Step 1]、[Step2]で共通部分列 "A → B → C → D", "A → D" が得られる(第1文字列のAと第2文字列の後ろのAが対応付けられた場合が "A → D")。よって、"A → B → C → D" が最長共通部分列(LCS)となり、残りの要素が差分となる。

なお〔単語−文字列〕間比較を行うため、Heckel法、LCS法とも[Step1]では、予め旧文書の各単語を単語の先頭文字に対応したハッシュ表に格納する。新文書の文字列を辿り、一致する単語が存在するかを調べる際に、その文字に対応するハッシュ表のみを検索することで、高速に一致する〔単語−文字列〕の組合せを見つける。

### 3.3 従来方式の検討およびその高速化手法

Heckel法およびLCS法により、〔単語−文字列〕間で差分抽出を行った計算時間と抽出精度を表1に示す。実験は6件の日本語文書データを対象とし、HP9000/750(約76 MIPS, 主メモリ96M)を用いた。なお抽出精度は、人手によって対応させた文字列の文字数のうち、正しく抽出された文字数の割合で示した。

表1 Heckel法とLCS法による実験結果

| データ | 文字数 [byte] | 計算時間 [s] | | 抽出精度 [%] | |
|---|---|---|---|---|---|
| | | Heckel | LCS | Heckel | LCS |
| data1 | 11,145 | 0.12 | 3.60 | 78.8 | 99.7 |
| data2 | 9,275 | 0.10 | 2.70 | 98.2 | 100.0 |
| data3 | 27,740 | 0.26 | 1001* | 99.6 | 100.0 |
| data4 | 8,551 | 0.09 | 2.20 | 99.6 | 99.6 |
| data5 | 4,342 | 0.06 | 0.62 | 98.5 | 98.8 |
| data6 | 10,847 | 0.11 | 2.50 | 88.0 | 98.8 |
| | | | 平均 | 93.8 | 99.5 |
| | | | 最低値 | 78.8 | 98.8 |

注*) メモリ消費量が大きく、メモリのスワップが発生している

表1よりそれぞれの方式の特徴をまとめると以下のようになる。

(1) 抽出精度はHeckel法がLCS法に比べて低い
これは、Heckel法は唯一の要素をkeyとして文書間の比較を行うため、唯一要素が少ないと一致文字列の抽出もれが起きるためである。LCS法は必ず最長に一致する文字列が抽出される。

(2) 計算時間はHeckel法がLCS法に比べて小さい
LCS法はすべての一致する要素の組合せをみるため計算時間、メモリ消費量ともにHeckel法に比べて一般に大きい。特に単語単位の比較は、行単位での比較等と比べて比較要素数が増えるため、〔単語−文字列〕間LCS法をそのまま用いると計算量が大きくなり、実用上問題があると考えられる。

そこで、両方式の長所を活かした、次のような2段階差分抽出方式を検討した。

**第1段階:** 〔単語−文字列〕間Heckel法により差分抽出を行う。

**第2段階:** 得られた差分結果で変更(挿入、削除、変更のうち)と抽出された文字列には、抽出もれが起こっている可能性がある。この変更と抽出された文字列に対して、〔単語−文字列〕間LCS法により、改めて差分抽出を行う。

### 3.4 2段階差分抽出方式の評価実験

実験結果を表2に示す。

表2 2段階差分抽出方式の実験結果

| データ | 文字数 [byte] | 2段階差分抽出方式 | |
|---|---|---|---|
| | | 計算時間 [s] | 抽出精度 [%] |
| data1 | 11,145 | 0.35 | 99.7 |
| data2 | 9,275 | 0.11 | 100.0 |
| data3 | 27,740 | 0.27 | 100.0 |
| data4 | 8,551 | 0.09 | 99.6 |
| data5 | 4,342 | 0.06 | 98.8 |
| data6 | 10,847 | 0.16 | 98.2 |
| | | 平均 | 99.4 |
| | | 最低値 | 98.8 |

表2より、Heckel法とLCS法を組み合わせた2段階差分抽出方式の場合、平均抽出精度99.4%、計算時間0.1 ～ 0.3[s]である。2段階差分抽出方式を用いることにより、LCS法を単独で用い最長に一致するものを抽出した場合(表1)に比べて、抽出精度を劣化させることなく、10分の1以下の計算時間での抽出を可能にした。

## 4. おわりに

本稿では、差分抽出時に単語分割することなく、迅速に単語単位の差分を抽出できる〔単語−文字列〕間差分抽出方式について述べた。また、第一段階として計算速度の速いHeckel法により、ある程度抽出範囲を絞り、第二段階でLCS法を用いて正確に抽出を行う2段階差分抽出方式が単語単位の差分抽出方式として適していることを示した。

## 参考文献

[1] 角田,"ファイル間の相違検査法", 情報処理, 24, 4, pp.514-520(1983)