日本語テキスト処理支援のためのＣライブラリの設計と実現

1Ｊ－2

天野純一，早川栄一，並木美太郎，高橋延匡　（東京農工大学）

１．はじめに

テキストデータの作成や閲覧を行なうアプリケーションプログラム（AP）のユーザは、検索などのテキスト処理を「語」に対して行なうことが多い。しかし日本語テキストの中の語を認識して処理する AP を作成することは難しい。

本稿では、この問題に対処するために語を単位として日本語の文章に対するテキスト処理を行なうＣライブラリについて述べる。

２．日本語テキスト処理の問題点と現状

２．１　語単位テキスト処理の必要性

筆者の研究室では現在、文章校正支援システムや、文章中の語の用例索引（コンコーダンス）を作成するシステムなどを研究している。文章を扱うこれらの AP では、ユーザは文字列というよりも語を意識している。しかし、日本語では AP は語の区切りがわからない。そのために、たとえば検索処理ではユーザが期待しないマッチング（「程」が「程度」に、など）が起きることがある。また、語の読みを必要とする処理は通常のテキスト処理では行なえない。

２．２　語の認識手段

文中の語の認識手段である形態素解析には多くの実現例があるが、解析処理だけでシステムが閉じていることが多い。形態素解析をライブラリ化した例[1]もあるが、多くの AP が必要としているテキスト処理は用意されていない。

３．作成目的

前章で述べた理由から、日本語テキストを扱う AP には、文字列ではなく「語」を単位にして検索・置換などの各種のテキスト処理を行なう機能を持つことが望ましい。しかし形態素解析は AP にとって扱いにくい。そこで、これを内部で利用して語を単位にしたテキスト処理を行なう機能を、AP 記述言語として広く用いられているＣのライブラリとして作成する。

４．設計方針

本ライブラリの設計方針を次に挙げる。

(1) 基本的な機能で構成する

AP に対する汎用性を重視して、複数の基本的なテキスト処理機能から構成する。

(2) 簡潔な API を提供する

AP が直接扱うライブラリ関数をプログラマに理解しやすい簡潔な形にする。

(3) 主記憶の消費量を抑える

AP が用いる主記憶領域を圧迫しないために、解析木全体の主記憶上での保持は行なわない。

(4) 拡張性を重視する

AP からの要求に伴うライブラリの拡張を後から行なうことを考慮した全体構成をとる。

５．設計

５．１　実現する機能と全体構成

実現する主な機能を次に挙げる。これらとAPおよび形態素解析の関係を図１に示す。また、これらの API を表１に示す。

(a) 形態素解析の実行を受け付ける機能
(b) テキスト中の語の位置を求める検索機能
(c) テキスト中の語の表記を置換する機能
(d) 語を読みの五十音順でソートする機能

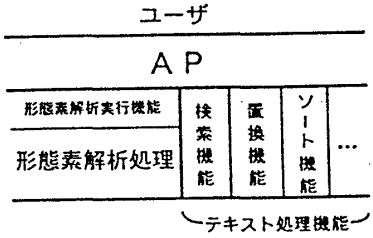

図１　ライブラリの全体構成

機能	ライブラリ関数の形式
解析	文章型 *文章オープン(CHAR *テキスト); 解析された文章
検索	LONG 語検索(文章型 *文章, 語型 検索キー, LONG 開始位置); 検出した語の位置　処理対象の文章
置換	INT 語置換(文章型 *文章, LONG 位置, CHAR *表記); 置換の成功／失敗　置き換える表記
ソート	INT 語ソート(語型 *語, INT 個数); ソートする語の配列

表１　ライブラリ関数の API

Design and Implementation of a C Library for the Support of Japanese Text Processing
AMANO Junichi , HAYAKAWA Eiichi , NAMIKI Mitarou and TAKAHASHI Nobumasa
Department of Computer Science , Faculty of Technology , Tokyo University of Agriculture and Technology

5．2　形態素解析の実行機能

テキスト処理のたびに形態素解析をしては効率が悪い。本機能はこれから処理したいテキストを AP から受け取り、句点で文を切り出して解析と探索を行ない、語を得る。探索結果は文単位で切れているが、テキスト処理機能には線形に連続した語に見えるように抽象化し（図２）「文章型」と定義した一つの変数として返す。

本機能をテキスト処理機能から分けることによって、解析の実行は一度だけですむ。また、テキスト処理機能の拡張を形態素解析に手を加えずに行なえる。

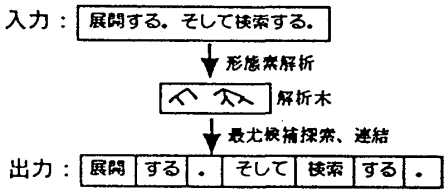

図２　ライブラリ中での解析処理

5．3　語の検索機能

コンコーダンス作成システムでコンコーダンス項目になる語を文章中から探すなど、検索処理はもっとも広く用いられるテキスト処理機能である。本機能は AP から検索キーと解析済みの文章を受け取り、キーにマッチする語の位置を返す。検索キーは語の表記の他に品詞を同時に指定することもできる。したがって、前後の字句との接続条件で語を指定する正規表現などの手法とは異なり、名詞「借りがある」と動詞「借りている」のような同表記異義語を区別することができる。本機能を用いたAP の記述例として、送り仮名の規則に合わない語の位置を検出する関数を図３に示す。

```
LONG    誤った表記の検出(文章型 *文章, LONG 検出開始位置)
{
        LONG    検出した位置;
        語型    検索キー;

        検索キー.表記 = "割り込み";
        検索キー.品詞 = 普通名詞;

        検出した位置 = 語検索(文章, 検索キー, 検出開始位置);
        return  検出した位置;
}
```

図３　検索機能の利用例

5．4　語の置換機能

ワードプロセッサにおける当て字や送り仮名の修正などには置換機能が用いられる。本機能は AP から文章と置換対象の語の位置、および置換後の表記を受け取る。そしてその語の表記を置換し、その結果は受け取った文章に反映される。

日本語表記の基準によっては、名詞は「割込み」動詞は「割り込み」と書くなど、品詞によって用いるべき送り仮名が異なる場合がある。これに対しては、図４のように検索機能であらかじめ品詞検索することによって対応できる。

```
extern  LONG 誤った表記の検出(文章型 *, LONG);
                /* 図4で定義した関数 */

VOID    誤った表記の修正(文章型 *文章)
{
        LONG    位置 = 0L;

        while((位置 = 誤った表記の検出(文章, 位置)) != -1L)
                語置換(文章, 位置, "割込み");
}
```

図４　品詞に対応した置換の例

5．5　語のソート機能

多くの語を一度に表示するような場合は五十音順に並べればユーザが確認しやすい。本機能は内部で辞書検索を行ない、漢字仮名変換をすることによって、AP から与えられた複数の語をそれらの読みの五十音順にクイックソートする。

6．実現

本ライブラリは筆者らの研究室で開発された形態素解析[2]を利用している。実現環境は、これも筆者らの研究室で開発し多くのテキスト処理 AP が動作している「OS/omicron 第２版」とＣコンパイラ「CAT」を用いている。現在までにソート機能と形態素解析実行機能を、それぞれ約 800 行および 300 行の規模で実現している。

7．おわりに

本稿では、語を単位にすることによって AP の日本語テキスト処理を支援するＣライブラリについて述べた。今後は完全な実現を行ない、各種の AP に適用していく。

参考文献

[1] 小松：日本語解析用クラスライブラリ「Jpn クラスライブラリ」の開発，第 48 回情学全大，4Q-5（1994）

[2] 下村，他：最小コストパス探索モデルの形態素解析に基づく日本文誤り検出の一方式，情学論，Vol.33，No.4（1992）

[3] 牛島，他：日本語文章推敲支援ツール『推敲』のプロトタイピング，コンピュータソフトウェア，Vol.3，No.1（1986）