テキスト・データベース管理システム SIGMA による日本語データの検索*

許 斐 慧 二

0．はじめに

近年，コンピュータ技術の著しい進歩によって，個人でも大規模なコーパス（機械可読の言語資料）が取り扱えるようになり，英語の研究者の間でコーパスを利用した研究に対する関心が高まってきている。言うまでもなく，コーパスを用いて言語研究をおこなう際に無くてはならないのがデータ検索のためのプログラムである。筆者はここ数年来データの検索には専らテキスト・データベース管理システムＳＩＧＭＡを用いており，前稿(1994,1995)において実際にこのシステムを使った英語の語法研究を紹介した。ＳＩＧＭＡは，英語ばかりでなく日本語のテキストにも対応できる。しかし，日本語のデータ検索では英語とは違った作業を必要とする場合がある。本稿では，まず初めに筆者が現在作成している日本語テキスト・ファイルの形式について簡単に述べ，次に日本語の否定対極表現の一つである係助詞「しか」を例にとって，ＳＩＧＭＡの使用方法を具体的に説明する。

1．日本語テキスト・ファイルの形式

言語研究者にとってＳＩＧＭＡシステムの大きな利点の一つは処理するファイルの形式に制限がないことである。しかし，このシステムの特徴を最大限にいかした検索を実現するには，ファイルもそれに合った形式にするのが望ましい。そこで，前稿(1994, 1995)では，英語のテキスト・ファイルを以下のよう

な形式に統一した。

(a) 復帰改行を（1行ごとでなく）1文ごとに行う。こうすることで復帰改行をレコード区切り語として使い，文単位でデータ検索を行うことができる。ただし，コロンやセミコロンのすぐ後では復帰改行しない。これらの記号の前後の文は合わせて1文とする。語法研究ではコロンやセミコロンの前後の文にまたがって検索しなければならない場合があるからである。

(b) 単語と単語あるいは単語と句読点や引用符号などの記号との間に1スペース分の空白を設ける。このように単語が必ず1スペース分の空白で区切られているようにすることで特定の単語だけを簡単に切り出すことができる。

(c) 各文の先頭に出典情報を含むインデックスを付ける。

(d) 各単語にタグは付けない。

以上から分かるように，筆者の作成した英語コーパスのテキストは，原則的に1文を1レコードとし，さらに，単語を切り出し易くするために多少手が加えられてはいるものの，基本的にはいわゆるプレーン・テキストである。

日本語のテキスト・ファイルもほぼ同じ形式にしている。ただ，(b)の措置は日本語には馴染まない。日本語には，大雑把な言い方をすれば，単語と単語あるいは単語と助詞などとの間に，スペースを置く習慣がないからである。もちろん，テキストを入力する際に，単語と単語，単語と助詞，あるいは単語と句読点などの記号との間に1スペース分の空白を置くことはできる。しかし，それは日本語文の表記としては不自然であるし，テキストの入力に手間もかかる。

日本語文に切れ目がないとすると，場合によっては，英語と同一の方法では文字列の切り出しができないことになる。しかし，実は，これはＳＩＧＭＡに

よる検索においては，さほど大きな問題とならない。後で見るように，このシステムに備わっている演算機能をうまく利用すれば，何ら手の加えられていないプレーン・テキストであっても，かなり正確なデータの検索が可能であるからである。

ここで，筆者が現在作成している日本語テキスト・ファイルのデータの形式を示しておきたい。

(1) KJF1MS02:041:10 @ ^ 彼女との関係はおおっぴらにしていないが，渡瀬の友人の何人かに連れ立っているところを見られている。

冒頭のインデックスは出典を表わす。'KJ'はKIT Corporaの日本語コーパス(KIT Japanese Corpus:以下，KJ コーパスと略称する)であることを示す。続く'F1'は資料の種類（'F1'はフィクション／小説である）を示す。'MS'は著者名の略字，02は作品番号である。2番目の数字列はページ番号を，3番目の数字列は文番号（ページの先頭から数えた文の番号）を示している。また，'@'はパラグラフの先頭を，'^'は文の先頭をそれぞれ表わしている。

日本語のテキスト・ファイルを作成する際にやっかいなのは，引用文の取扱いである。次の文を見てみよう。

(2) 「彼女，私のこと警戒しているみたい。私から電話があったと知れば備えを固めるかもしれないわ。警察はすでに蔵方夫人を疑っているようだったわ。石の同一性を証明できればかなり網を絞り込めるんじゃないかしら」

(2)では，1つの引用部に4つの文が生じている。こうした引用部の場合に，いくつの文を1レコードとするかはケース・バイ・ケースであるが，たいていの場合は1文1レコードとし，末尾や先頭に'>'を置いて，後続の文や先行する

文が存在することを示すようにしている。[1] '>' と '>' に挟まれた文は3つ以上の文から成る引用部で先頭と末尾の文の間に位置する文である。例えば，(2)の文は次のように表示される。

(3) KJF1MS01:012:12 ^ 「彼女，私のこと警戒しているみたい。>
KJF1MS01:012:13 ^ >私から電話があったと知れば備えを固めるかもしれないわ。>
KJF1MS01:012:14 ^ >警察はすでに蔵方夫人を疑っているようだったわ。>
KJF1MS01:012:15 ^ >石の同一性を証明できればかなり網を絞り込めるんじゃないかしら」

特に問題なのは，(4)のように，引用部が文の中間に生じている場合である。

(4) 弁護団長の渡辺脩弁護士は，「法廷での議論や弁論方法についての批判は自由だと思う。けど，弁護活動そのもの，弁護人になることを非難しないでほしい」と訴えた。

もし1文1レコードの原則にしたがって，文中に生じた引用文をも1レコードとして取り扱おうとすると，引用文は主節の部分と分けてたてなければならず，節の主従関係を表示する，何らかの方法を考案しなければならないだろう。筆者は，このような場合には，検索のしやすさやデータの見やすさなどを考えて，文全体を1レコードとして扱っている。（ＳＩＧＭＡではレコード区切り語で挟まれた部分が検索単位の1レコードとみなされる。筆者は，復帰改行をレコード区切り語として用いているので，どのような文字列であれ，入力の際に末尾で復帰改行を行いさえすれば，1レコードとして取り扱うことができる。）

正確なデータ数の把握が必要とされる統計的な研究では，ここで示したよう

な検索単位の設定の仕方には問題があるかも知れないが，筆者が日本語コーパスを利用する主たる目的は，理論的な研究において内省による作例の不足を補うため，あるいは，提案された仮説を検証するためであるから，これで差し支えないと考えている。

2．ＳＩＧＭＡによる検索

では，ＳＩＧＭＡを使った日本語のデータ検索の具体例を見てみよう。単語あるいは連続した語句の検索では，多くの場合，その検索したい単語あるいは語句をキーワードとして登録してやればよい。例えば，「警察は泥棒が逃げていくところを捕まえた」のように「ところを」という表現を含む例文を抽出したい時には，「ところを」をキーワード登録すればよい。また，不連続な文字列の場合でも，その形が定まっているものなら，トリプル・ドットを用いて，検索を行うことができる。例えば，係り結びの「こそ～けれ」を検索するには，'こそ...けれ'とキーワード登録してやればよい。こうしたケースについては，前稿(1994，1995)で紹介した英語データの検索方法がそのまま利用できるので，そちらを参照していただきたい。本稿では，検索する際に日本語に特有の工夫を必要とするものとして，いわゆる「しか～ない」構文を取り上げることにする。

なお，検索に用いるKJコーパスの現在のサイズは約13.65メガバイトで，その内訳は小説が1.38メガバイト，エッセーが0.72メガバイト，科学的読物が0.1メガバイト，週刊誌の記事が0.95メガバイト，それに新聞記事が10.5メガバイトである。

さて，係助詞「しか」はいわゆる否定対極表現の一つで，典型的な例では，常に，同じ節の中に否定辞「ない」，あるいは，その活用形を伴う。[2] この係助詞「しか」と否定形のペアは英語の連語と似ているので，すぐ思い付く一番

簡単な検索方法は，英語の‘prevent ～（from）～ing’の検索の場合と同じように，キーワードとして，「しか」と「ない」あるいは「ない」の活用形を組み合わせて（つまり，‘しか...ない’，‘しか...なかった’などのようにトリプル・ドットを用いて）登録することである。しかし，この方法では，検索対象の「しか」を含むデータを取りこぼしてしまう可能性がある。「しか」が共起する否定表現のすべてをキーワードとして登録できるとは限らないからである。結局，否定形は検索対象とせず，「しか」をキーワードとして検索するのが最も安全な方法ということになろう。

だが，「しか」だけをキーワードとして検索すると，例えば，「しかし」，「もしか」などのように「しか」という文字連鎖をその一部とする表現を含む，不要なデータがすべて取り出されてしまう可能性がある。そこで，論理演算子‘and’，‘or’，‘not’を用いて，(5)の論理式を定義してみる。

(5) Keywords

A１：しか

A２：しかし

A３：もしか

Logical Formulas

Z１：A１.^(A２,A３)

これによって，係助詞の「しか」は含み，これと無関係な「しかし」や「もしか」は含まないデータが検索できる。しかし，論理演算子に基づいた(5)のような論理式では，「しかし，太郎しか来なかった」などのように，検索対象の「しか」と排除されるべき表現「しかし」とが両方含まれる文は取り出せない。

ここで注目すべきは，この文では「しか」は２回出現しているのに対して，「しかし」は１回であるという点である。こういう場合には，ＳＩＧＭＡのキー

ワードの出現回数をカウントする機能が役立つ。(6)はこの機能を用いて定義された論理式である。

(6) Z1：A1-A2-A3

しかし，(6)の論理式でも，係助詞「しか」の検索にはまだ不十分で，「もしかしたら，太郎しか来ないかも知れない」のような文は取り出せない。なぜなら，「もしかしたら」には「しか」，「しかし」，「もしか」の３つが合致するので，カウントが-１（'1-1-1=-1'）であり，後に続く「太郎しか」の「しか」を加えても，文全体としてのカウントは０（'2-1-1＝0'）にしかならないからである。この文を間違いなく取り出すためには，「もしか」と「しかし」を重複させた「もしかし」（これで「もしかして」と「もしかしたら」をカバーできる）を足し戻す必要がある。[3] それを定義した論理式が(7)である。

(7) Keywords

A1：しか

A2：しかし

A3：もしか

A4：もしかし

Logical Formulas

Z1:A1-A2-A3+A4

以上，ごく簡単に，どのようにしたらＳＩＧＭＡのキーワードの出現回数をカウントする機能を利用して検索対象の文字列と同じものをその一部に含む文字列を除くことができるか示した。係助詞「しか」の実際の検索では，もっと多くのキーワードの登録とその引き算，足し算が必要である。以下に，その１例を示す。なお，ＳＩＧＭＡによる検索で用いられるコマンドの解説やレコー

ド区切り語などの指定の仕方，キーワードの登録の仕方などに関する詳細な説明は前稿(1994, 1995)でおこなっているので，本稿では簡単に述べるにとどめる。

(8) A:¥WRK>search[R]

Record Delimiters

R 1 =¥n[R]

R 2 =[R]

Item Delimiters

I 1 =[R]

(9) Keywords

A 1 :=しか[R]

A 2 :=しかし[R]

A 3 :=しかも[R]

A 4 :=しかった[R]

A 5 :=たしか[R]

A 6 :=しかけ[R]

A 7 :=もしか[R]

A 8 :=しかめ[R]

A 9 :=しから[R]

A10:=しかね[R]

A11:=しかる[R]

A12:=しかたな[R]

A13:=しかたがない[R]

A14:=こころなしか[R]

A15:＝心なしか[R]

A16:＝しかしながら[R]

A17:＝おしかり[R]

A18:＝痛しかゆし[R]

A19:＝のしかた[R]

A20:＝にしかず[R]

A21:＝なにがしか[R]

A22:＝何がしか[R]

A23:＝しかつめらしい[R]

A24:＝いつしか[R]

A25:＝のしかか[R]

A26:＝さしかか[R]

A27:＝しかしら[R]

A28:＝しかして[R]

A29:＝しかしな[R]

A30:＝もしかし[R]

A31:＝たしかめ[R]

A32:＝たしから[R]

A33:＝[R]

(10) Logical Formulas

Z1:＝A1

Z2:＝A1-A2-A3-A4-A5-A6-A7-A8-A9-A10-A11-A12-A13-A14-A15-A16-A17-A18-A19-A20-A21-A22-A23-A24-A25-A26+A27+A28+A29+A30+A31+A32[R]

Z3:＝[R]

(11) Input File:=¥KJ¥*.*[R]

Searching ¥KJ¥KJF1MY01.TXT ...

.

.

.

RETRIEVED RECORDS

QUESTION 1(Z1) =188 3082

QUESTION 2(Z2) = 38 603

TOTAL =188 3082

CPU TIME (second) = 9 123

(12) A:¥WRK>refile[R]

RETRIEVED RECORDS

QUESTION 1 (Z 1) = 3082

QUESTION 2 (Z 2) = 603

TOTAL = 3082

Question Number:=1[R]

New Record Delimiter:=¥n¥n[R]

Output File:=sika-1[R]

Question Number:=2[R]

New Record Delimiter:=¥n¥n[R]

Output File:=sika-2[R]

(検索方法の解説)

(8) search コマンドの入力，レコード区切り語の指定及び項目区切り語の

指定。[R]は復帰改行の入力（リターン・キーの打鍵）を表わす。

(9) キーワードの登録。キーワード変数A１は「しか」を，Ａ２～A26はA1によって検索されるものから排除すべき「しか」を含む文字列を，A27～A32は足し戻すべき「しか」を含む文字列を登録している。[4]（例えば，「しかし」を検索すべき対象から排除すると，「それは太郎しかしなかった」などの文は抽出できないことになる。そこで，「しかしな」を足し戻すべき文字列としてA29に登録している。）

(10) 論理式の登録。論理式Z1によって「しか」を含むレコードがすべて検索される。論理式Z2によって，検索対象でない「しか」をその一部とする文字列は含まず，「しかしな」などの検索対象となる文字列は含む，「しか」のレコードが検索される。検索対象の係助詞「しか」を含むレコードはこの論理式によって検索されるものの中に含まれる。

(11) 入力ファイルの指定。

(12) refile コマンドの入力及び検索結果の再ファイル化。

3．検索結果の検討

上の検索結果から分かるように，KJコーパスには「しか」を含む用例が3082個見られる。このうち，係助詞「しか」を含む可能性のある用例は603個である。そこで，エディターの検索機能を用いて，その603個を一つ一つ点検してみると，実際に係助詞の「しか」を含むのは561例であった。念のために，「しか」の全用例3082個を，これも同じ方法で点検してみると，係助詞「しか」を含む用例はやはり561個であった。上に示した検索方法で，必要なデータは確実に抽出できたことが分かる。

ところで，上に示したキーワードの登録の仕方では42個の不必要なデータを排除することができなかった。さらに検索の精度を高めるには，これらのデー

タの中身をよく吟味して，より効率的なキーワードの登録の仕方を考えればよい。しかし，筆者は，上で示した以上の努力は無駄であると考えている。なぜなら，検索対象と同一の文字列をその一部に含む文字列が数多く存在する場合には，必要なデータのみを取り出すことは殆ど不可能であり，いずれにせよ研究者が自分自身の眼で検索結果を点検しなければならないからである。検索の際にある程度の量のゴミが含まれるのはやむを得ない。むしろ，必要なデータが抜け落ちないようにすることの方が重要である。

4．おわりに

以上，筆者が現在作成中の日本語コーパスを用いて，テキスト・データベース管理システムＳＩＧＭＡの使用方法を具体的に説明した。日本語はその表記法のゆえに単語の切り出しが難しく，データの検索には英語の場合とは違った困難が伴う。しかし，本稿で見たように，ＳＩＧＭＡの演算機能をうまく利用すれば，かなり正確に検索対象を抽出することができる。ＳＩＧＭＡは，データを検索する際に，研究者が自分でキーワードを登録し，かつ，論理式を定めなければならないために，一見とっつきにくく感じられるが，その演算機能をよく理解し，少し使い方に慣れれば，これほど便利で力強いデータ検索用のプログラムはないように思われる。

註

＊テキスト・データベース管理システムＳＩＧＭＡによるデータ検索の方法について今回も懇切に御教示くださった九州工業大学情報工学部知能情報工学科の篠原武教授にまずお礼を申し上げたい。同学科の為近光宏氏（現，奈良先端科学技術大学院生）には，日本語のデータ検索の方法をお教えいただいただけでなく，英語と日本語の両方に対応できるデータベース作成プログラムと日本語のデータ検索でキーワード入力の手間を省く検索用質問式作成支援プログラム(GENKEY)を開発していただいた。また，楠木美雪さんには日本語テキスト・ファイルの作成をお手伝い

いただいた。ここに記して感謝の意を表したい。

1．引用部が2文のみの場合には，その2文を一つのレコードとして扱っている。また，レコードの設定にあたっては引用部がいわゆる短文（minor sentence）を含むかどうかなどの要因も考慮に入れている。
2．係助詞の「しか」と共起する否定表現には「ない」とその活用形以外に，やはり否定を表わす「～ず」，「駄目だ」，さらに反語表現の「～ものか」などがある。「しか」の付与された句と否定表現との間には何らかの統語的関係が成立していなければならないことが知られている。近年，生成文法においては否定対極表現の認可の問題が注目を集め，盛んに議論されている。ここではそうした理論的な問題には触れない。許斐(1989)，Konomi(1996)を参照されたい。
3．検索の際に排除したい語をネガティブ・キーワードと呼ぶ。「しかし」や「もしか」はネガティブ・キーワードである。2つのネガティブ・キーワードに引っかかるものはもう一度足し戻す。そうした追加される語をポジティブ・キーワードという。「もしかし」はポジティブ・キーワードである。
4．ＳＩＧＭＡシステムの search を用いた検索では，このように，正しく検索対象を抽出するために，検索対象でない文字列もキーワード登録しなければならない。キーワードとして登録しなければならない文字列の数は，ここに示した係助詞「しか」の検索例からも分かるように，かなり膨大になる。謝辞でも紹介したように，為近光宏氏はこうしたキーワード入力の労力を軽減するために検索用質問式作成支援プログラム GENKEY（generate keywordsか？）を作成しているが，ここでは，同プログラムには頼らず，ＳＩＧＭＡ自体の演算機能を利用して検索している。

参考文献

1．有川節夫，篠原武ほか．1987．テキスト・データベース管理システムＳＩＧＭＡ第2版について．『九州大学大型計算機センター広報』Vol.20, No.6, pp517－581.

2．川根友恵．1991．『文字列パターン照合に基づく英文例文検索に関する研究』九州工業大学情報工学部卒業論文.

3．許斐慧二．1989．「しか～ない」構文の構造．『英語学の視点』pp369－392．九州大学出版会.

4．＿＿＿＿．1994．テキスト・データベース管理システムＳＩＧＭＡを用いた語法

研究．『九州工業大学情報工学部紀要』（人文・社会科学）第７号，pp65－87．

5．_____．1995．テキスト・データベース管理システムＳＩＧＭＡによるデータ検索．田島松二編著『コンピューター・コーパス利用による現代英米語法研究』pp205－226．開文社出版．

6．Konomi, Keiji. 1996. On Licensing of SIKA-NPIs in Japanese.『九州工業大学情報工学部紀要』（人文・社会科学）第９号，pp83－121．

7．為近光宏．1996．『文字列パターン照合を使った日本語テキストからの例文検索』九州工業大学情報工学部卒業論文．