古典文の総索引を用いた品詞タグ付きコーパスの自動作成

矢古宇 智子*　潮 靖之**　上原 徹三*
*武蔵工業大学工学部　**武蔵工業大学大学院工学研究科

1 目的

日本語研究においてコーパスを利用する場合が多々ある．しかし，古典文献については品詞付けされた電子化コーパスは無いと思われる．そこで，本研究では市販されている電子化総索引および本文を利用し，その漢字表記や品詞情報を活用することで品詞タグつき古文コーパスの自動作成をする．しかし実際には，完全自動作成は不可能であり，半自動作成後に人手で訂正して，完成させる．総索引とは，本文のすべての単語について，品詞・活用形および本文の出現位置を記したものである。この索引を行ごとに集めて再構築すれば本文を再現でき，コンピューター処理でのコーパス作成を可能にしている．

対象の総索引は歌物語 [1]（「伊勢物語」「平中物語」「大和物語」），日記文学 [2]（「土佐日記」「蜻蛉日記」「和泉式部日記」「紫式部日記」「更級日記」）である．

2 総索引の内容と形式

2.1 索引部分

索引部分は五十音順に並んだ仮名表記（活用語の場合はその終止形）の語を見出しとして，漢字表記，品詞，各活用形が現れる行番号等が示されている．それ以外にも音便（イ音便，ツ音便など）がある場合などは，行番号の後ろに「イ」や「ツ」などの音便記号を表記する．文献中の簡略化した例を以下に示す．

＜例＞大和,618,あかしのうら,明石浦,AO

上記の内容としては文頭から順に，物語名 (大和物語)，行番号（６１８）仮名表記 (あかしのうら)，漢字表記 (明石浦)，品詞と活用 (名詞) の各要素となっている．　索引には文献中の全語が収録され，以下のような規則によって表記されている．

- 見出し語は，著者が一語と判断した範囲で記載されている．この中には複合語，人名なども含んでいる．
- 「名詞＋す（サ変動詞）」「動詞連用形＋す」などは，「す」までを一続きとする．
  ＜例＞土佐,444,,ふなゑひす,船酔,D2,,,,K
- 活用語については，原則として終止形を見出し語とする．
- 漢字表記については，送りがなを含まない．
  ＜例＞土佐,23,,よぶ,呼,12,,,,

(注：索引中の漢字表記と本文中の漢字表記は，必ずしも一致しない．また漢字表記があっても，本文は仮名である場合もある)

2.2 本文部分

文献中の例を以下に示す．
＜例＞ 0028,002, のちまで侍ひける。
上記は文頭より行番号（００２８），段番号（００２），本文を表す．本文部分に用いられている規則を以下に示す．

- 歌物語の本文においては、索引の見出しとなる単語は、本文の二行にまたがることはない。日記物においては例外もある．

2.3 古文コーパスの形式

作成する古文コーパスは，EDR 日本語コーパスの形式仕様をほとんどそのまま使用することとする．1 レコードの構造は，例文番号，出典文献，本文，表記，読み，品詞，活用情報とする．古文コーパスの１レコード作成例を以下に示す．

```
HC00000037 土佐日記　！　浦戸より漕ぎ出でて、大湊を追ふ。　｛　1　浦戸　うらと　名詞　＊　2　よ
り　より　助詞　＊　3　漕ぎ出で　こぎいで　下二段動詞　連用　4　て　て　助詞　＊　5　、　、　記　号　＊　6　大湊　おほみなと　名詞　＊　7　を　を　助詞　＊　8　追ふ　おふ　四段動詞　終止　9　。　。　記号　＊　｝　/1:浦戸 /2:より /3:漕ぎ出で /4:て /5:、 /6:大湊 /7:を /8:追ふ /9:。 /　( ) [ ] DATE=" "
```

上記の例では文頭から順に，例文番号（００００００３７），出典 (土佐日記)，本文，各単語と品詞の情報：括弧から括弧の間，形態素情報：/1 から/9 まで，という内容になっている．

3 実現方法

上記に述べたとおり，例文と読み，品詞情報よりなるＥＤＲ形式コーパスを，前述の電子化索引および電子化本文を用いて作成する．そのおおまかな処理の流れを以下に示す．なお処理は行単位を基本とし，処理を行ごとに繰り返すことで一つの作品についてのコーパスファ

イルを作成する. 第1段階のファイルが1作品分出来上がったら, 手直しを加え, 最終的にコーパス表記を作る第2段階のプログラムへ通すものとする. 以下に処理の概要を示す.

第1段階 .

(1) 本文の行番号より, 同じ行に該当する索引を取り出す.

(2) マッチングの際に必要となる各種情報を付加する. 主に見出し語を活用させて本文との照合が可能な形に変化させる.

(3) 本文と索引とを照らし合わせ, マッチングさせる. 本文の各文字に索引を照らし合わせていき, 同一と思われる部分に索引を当てはめていく.

(4) 決定した候補を確認する. 重複および欠損は再処理し, できる限り本文と対応する語を一意に確定するようにする.

(5) 不足情報 (”「”や句読点に関する情報) を付加し, 見やすく直しやすい形に整える.

(6) ファイルに第一段階の結果を出力する. これは行単位での出力とし, コンピューターの推定部分や推定, 失敗部分を記載したものとする.

手作業 第一段階での失敗点･不安点を手作業により直す.

第2段階

(1) 手直しした情報を元に, 本文にセットする.

(2) 本文と同じ形に並べられた索引を,EDR コーパス形式に整える.

(3) コーパスの形式として出力する.

以下, 上記の各処理の詳細を記す.

3.1 第1段階処理

本文の読み込みから, 単語の情報加工とマッチングまでの処理を第1段階処理とする. 単語の読み込み, 情報の分類, 本文の読み込みなどの前処理を行った後に, 重複単語の処理などの主要処理を行う. 以下にその処理の詳細を例と共に示す.

- マッチングの際に必要となる各種情報の付加
  索引の単語は終止形で記載されており, 活用形と音便との情報が記号によって与えられている. この記号分類に沿って索引の見出し語を活用・変換させ, 原文記載通りの元の形に戻す必要がある. プログラムにはすでに各活用に関しての変換法が入っている. 変換は仮名表記に対してのみ施される. 音便等も同様の処理となる.

  <例>土佐,242,, あし, 悪,S2,,,,
  S 2は, 形容詞シク活用：連用形を表しているので「あし」を「あしく」と活用させる.

次にマッチングである. 方法としては, 本文文字と索引の最初の文字を一文字ずつ照らし合わせていき, 合致した場合に, その文字を起点として2文字目以下のマッチングを行っていく. 本文と索引とのマッチングを取る際には, 漢字表記でのマッチングを最優先とし, 漢字のマッチングに失敗した場合に見出し表記による候補を選択するものとする. 漢字も見出し語もマッチングには最長一致法を使うこととする.

しかし, 先に述べたとおり, 漢字表記には送り仮名がつけられていない. そのため, 漢字マッチングの際には最長一致法とともに, 一文字先読みの方法で候補の決定をしていく. 以下にその処理の詳細を示す.

- マッチング手法1・最長一致法
  本文と索引とを照らし合わせ, 漢字候補の先頭文字と一文字ずつマッチングさせていく. 該当箇所が見つかった場合, 次の文字から後を照合し, マッチしているかどうか調べる. 漢字表記の該当候補が重複した場合, そのマッチング語数の一番多かったものを第1候補とする. 漢字候補のマッチングが失敗した場合, すなわち本文中に該当個所が見つからなかった場合, 見出し語により同じく最長一致法をおこなう.

- マッチング手法2・一文字先読み
  漢字表記の場合, 送り仮名は省かれている. これは漢字と漢字の間に入った仮名でも同じである. たとえば「菊の花」という単語の場合, 漢字表記は「菊花」となり, 間に入っていた仮名が抜かされている. この問題は地名や人名に多く見られ, 解決するためには一文字先読みの手法を用いる. これは一文字失敗したとしても, もう一文字先まで比較してから失敗かどうか決める方法である. 一文字先も該当漢字と合致しなかった場合, それまでマッチングした文字数をマッチング成功文字数として採用する. 図1に例を示す.

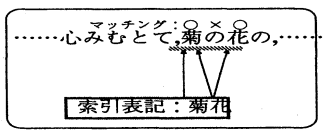

図 1: 先読み

以上の作業により索引を本文と同じ順番同じ表記に並べ, 索引には無い句読点の情報などを加えて第1段階として出力する.

3.2 第1段階処理の結果

出力された第1段階の結果は, 人手で直しやすいように文, 索引, 失敗箇所等が整えられて出力される. 以下にその例を示す.

<例>となむ、下りける日、いひやりける。

```
        *****?**********
--------------------------------
0番 * 0 * (<M>となむ 0 6 助詞 $)
1番 * 0 * (<M>、 6 2 記号 $)
2番 * 0 * (<K>下 8 2 四段動詞 連用)
3 番 * 0 * (<M>ける 12 4 助動詞 連体) (<M>け
る 28 4 助動詞 連体)
4 番 * 0 * (<K>日 16 2 名詞 $) (<M>ひ 22 2 名
詞 $)
5番 * 0 * (<M>、 18 2 記号 $)
6番 * 0 * (<M>いひやり 20 8 四段動詞 連用)
7 番 * 1 * (<M>ける 12 4 助動詞 連体) (<M>け
る 28 4 助動詞 連体)
8番 * 0 * (<M>。 32 2 記号 $)
--------------------------------
```

上の例の一行目は本文を示す. 2行目の?はその上の本文文字が候補無しの部分であることを示す. 4行目以下12行目まで(番号のついている行)は本文該当の索引とそのマッチング状況を示す. 次のような情報を含む.

- ＊0＊：何番目の候補が選択されたかを示す. この場合は0番目候補. ここにマイナス番号がついている場合は, マッチング失敗(重複もしくは欠損)であることを指している.
- KまたはM:Kが漢字表記を,Mが仮名表記を示す.

この第1段階での失敗箇所（二重該当や候補無し）に訂正を加えたファイルを第2処理に掛ける.

3.3 第2段階処理

第1段階の出力結果を手直しし, 第2段階の処理にかける. ここでは候補の大半はすでにマッチングされているため, 失敗例の箇所のみマッチングし直す. 直された文はコーパス形式の体裁に整えられ, 最終完成版として出力される. 出力形式は2.3で述べたEDR形式である.

4 処理における失敗および問題点

第1段階で出力された結果の, 見直しおよび訂正すべき箇所の例を以下に示す.

[1] <作成失敗>と表示された行

該当するはずの候補が本文のどこにもマッチングしなかった場合や, 同表記・同品詞・異活用や同表記・異品詞・異活用などによって候補が重複し, 本文との対応が一意に定められない場合, 作成失敗として処理され, 表示される.

次の例では2行目に作成失敗とあるので, 失敗文例であることが分る. 失敗した箇所は-1と示された部分である. 助動詞連用形の「て」と助詞の「て」の判別ができず, 候補を一意に定めることができない.

訂正の目安として＊と?のマークを使用する.?マークがついた文字は候補無し部分であり, そのまま処理すれば前の単語の送り仮名として処理される部分である. 送り仮名部分に＊がついている場合はあきらかに別単語とのマッチングであり, 単語の先頭語に?マークがついている場合は候補無しと判断できる. この場合, 候補の重複を人手によって振り分けたり, 欠けている単語を補ってやったりすることによって, 候補を完全に本文に対応させることとする.

<例>とどめてとりかへし給うてけり。

```
          ++++++++++<作成失敗>++++++++++
          ***?*******??***
----------+++++++++----------
62番 * 0 * (<M>けり 24 4 助動詞 終止)
63番 * 0 * (<K>給 18 2 補助動詞 連用)
64番 * -1 * (<M>て 6 2 助動詞 連用) (<M>
て 22 2 助動詞 連用)
65番 * -1 * (<M>て 6 2 助詞 *) (<M>て 22 2 助
詞 *)
66 番 * 0 * (<M>とどめ 0 6 下二段動詞 連
用) (<M>とどめ 8 2 下二段動詞 連用)
67 番 * 1 * (<M>とりかへし 0 2 四段動詞 連用
) (<M>とりかへし 8 10 四段動詞 連用)
68番 * 0 * (<M>。 28 2 記号 *)
----------+++++++++----------
```

[2] 成功文だが, 本文マッチング表示部分に?がついている場合

本文と候補とのマッチングを示す部分(上から2番目の***の行)に,?マークついた場合, 本文中のその部分は該当候補がない, ということを示している. 本研究で使用した索引には送り仮名が付けられていないため, 各単語の間に存在する候補不在部分は, コンピュータ処理では送り仮名と判断され, 自動的に各単語に割り振られている. 図2に例を示す. 大抵の場合その推測は正解

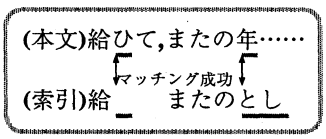

図2: 送り仮名

だが, 中には別候補と送り仮名部分が偶然にマッチングしてしまう場合もあるため, 成功文例でも?マークのある文は注意する必要がある.

訂正すべき箇所は多々あるが, 問題別に2つのパターンに分けることができる. コンピューターの処理過程の失敗と, 索引製作者の過失によるものとである. それぞれの詳細を以下に示す.

(タイプA) コンピューター処理上の問題による失敗

- 同語, 同品詞, 別活用の語が一文に複数ある場合
  <例>上一段動詞連用形の「見(送り仮名無し)」, 上一段動詞已然形の「見(送り仮名無し)」
- 同語, 別品詞の語が一文に複数ある場合
  <例>助詞の「に」と助動詞連用形の「に」

(タイプB) 索引上の問題による失敗

- 索引からその語が抜け落ちている

		伊勢	平中	大和	更級	紫	和泉	蜻蛉	土佐	合計
文	全数	550	439	768	650	501	337	1927	147	5319
	成功	432	308	385	450	166	197	454	25	2417
	%	78	70	50	70	33	58	24	17	45
	失敗	118	131	383	200	335	140	1473	122	2909
	%	27	30	50	30	67	41	75	83	55
行	全数(和歌無し)	1202	1027	1602	1593	1720	922	3057	552	11675
	チェック	651	587	833	942	1218	584	2029	340	7184
	%	54	57	52	60	71	63	51	62	62
形態素	全数	8007	8303	15414	11097	13849	7154	36061	4859	104744
	成功	7806	8104	14656	10750	13307	6873	33304	4615	99415
	%	97	98	95	97	96	96	92	95	95
	失敗	165	199	665	278	442	238	2457	195	4639
	%	2	2	4	2	3	3	7	4	4
	候補無し	36	20	93	69	100	43	300	49	710
	%	0.4	0.2	0.6	0.6	0.7	0.6	0.8	1.0	0.6

図 3: 結果

- 単語二つを連ねて作られている単語の場合
  ＜例＞言い言いて
  漢字表記は「言々」となっているため, 1文字目と3文字目の両方にマッチングしてしまう. どちらが正しいのか判別できず, 処理失敗
- 複合語が二行に分割されている
  ＜例＞花の賀の
  索引収録は「花賀」なのに, 本文では「花の」と「賀」の間で改行されている. 索引収録語は改行によって分割されないという規則のもと処理をしているので, 複合語が2行にわたるとそれを一語と認めることができず, 認識失敗となる.
- 漢字の送り仮名と一語の助詞とのミスマッチング
  ＜例＞ほどへて, 宮仕へする人なり
  下2段動詞「へ」は, 3文字目とマッチングされるはずであるが単語「みやづかへ」の漢字表記に送り仮名表記が無く「宮仕」となっているため, その中の「へ」ともマッチング成功してしまう. プログラムには漢字表記と本文表記の相違か判別できないため, マッチング失敗と処理される.
- 漢字表記が索引と異なる場合
  ＜例＞
  本文：0004,001, 里に、いとなまめいたる女はらから, ,
  索引：伊勢,4, をんなはらから,, 姉妹,A0,,,,,

これらの場合は機械での推定も及ばず, 第一段階を終え失敗処理となった時点で, 手作業により直すこととする. 上記のように, 第1段階の手直しには古語知識を必要とするものも多いため, 手直しをする人物にはそれなりの古語および古語文法に対する基礎知識が必要となってくる.

5 結果とその検討

第1段階の出力結果を図3に示す. なお, 行の”チェック”という欄は処理によって推測された？マークの部分を含む行の数である. 表から, 文成功率の平均は45%, 失敗率は55%であった. 最高は伊勢物語の78%, 最低は土佐日記の17%となっている. また形態素ごとの成功率は平均95%であり, 形態素単位ではほとんど成功だったことが分る. 蜻蛉日記をのぞいては, 失敗率も5%以下に抑えられている. このことから, 文のマッチング失敗の原因としては形態素のマッチング失敗よりも, 語の候補を一意に選べない場合が主であることが分った.

6 まとめ

以上で八作品のコーパスの作成は一応の完成をみた. 電子化された索引および本文を利用すれば, 人手作業よりも労少なく新規コーパス作成が可能だと分った. 今後, プログラムだけでなく索引形式もコンピュータ処理の観点で見直すことによって, より正確で手間のかからないコーパス作成が可能になると考えられる.

参考文献

[1] 西端 幸雄、木村 雅則共編(1994). 歌物語総合索引. 勉誠社刊行

[2] 西端 幸雄、木村 雅則、志甫由紀江編(1996). 平安日記文学総合索引. 勉誠社刊行