社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＳＬＰ－69 (54)
2007／12／21

日本語における略語自動生成法の検討とその音声インタフェースへの応用

榎 将功† 皇甫 美華† 大田 健紘† 柳田 益造†

† 同志社大学 〒610-0321 京田辺市多々羅都谷 1-3
E-mail: †dtg0702@mail4.doshisha.ac.jp

あらまし 音声認識に，認識対象として未登録の略語を使えるようにする方法を提案している．略語の生成はいくつかの規則に従うことが知られている．本研究では，それらの規則により元の表現（原型）から簡略後の表現（略語）を自動的に生成することを考えている．規則の適用により略語の候補を多数生成し，各候補に対し，どの規則を適用して生成したか，略語の言語モデルに整合しているか，Web 上での使用頻度は多いか，の 3 つの基準により略語らしさとしてのスコアをつけ，上位からいくつかの候補を選んで認識対象辞書に加えるという方略を提案している．提案法により原型 40 語から略語候補を生成し，各原型につき略語らしい候補を 10 語ずつ選んだところ，約 80%の略語をカバーできている．音声認識システムに提案法を応用したところ，認識語彙の増大による認識率の低下を十分上回る略語認識ができるようになっている．

キーワード 略語，生成規則，絞り込み，形態素解析，言語モデル，検索

Automatic Generation Abbriviated Forms of Japanese Expressions and its Apprications to Speech Recognition

Masanori ENOKI†, Mika KOHO†, Kenko OTA†, and Masuzo YANAGIDA†

† Doshisha University
1-3, Tatara-Miyakodani, Kyo-Tanabe, Kyoto 610-0321
E-mail: †dtg0702@mail4.doshisha.ac.jp

Abstract Proposed is a method to generate abbriviated forms of Japan expressions to accept them as words to be recognized even in case they are unregistered for speech recognition. It is known that there are several rules to generate abbriviated forms from original expressions. Proposed is automatic generation of abbriviated forms from an original expression. The proposed method generates several tens or hundreds of candidates of an abbriviated form by applying possible generation rules to the original expression. A scoring system to prune the candidates for each original expression is designed on the following three criteria; which generation rule is adopted, accordance with the language model of abbriviation, and appearance frequency on the Internet. Candidates having score ranked within the top N are registered into the word list for recognition. To evaluate the method, the proposed method is used to generate candidates of abbriviated forms from 40 original expressions, and the system choses 10 candidates for each original expression referring to the score. About 80% of the correct abbriviations were included in the top 10 candidates. The output of the proposed method is fed to a speech recognition system yielding recognition improvement sufficiently compensating decrease of recognition rate due to enlargement of vocabulary size.

Key words abbriviations, generation rules, pruning, morphological analysis, language model, retrieval

1. はじめに

増加し続ける情報に対応でき，かつ操作性の良いインタフェースの開発が要請されている．我々はそれに応えるために音声インタフェースの導入を検討し，その一例として，音

声入力によりテレビを操作するシステム(注1)（以降,「テレビ操作システム」と呼ぶ）を構築した[1]．その背景には，昨今の情報家電の高機能化や多機能化に伴う操作の複雑化があり，そのために操作上の困難が増大してきた．テレビの場合，番組の視聴予約や録画予約，オンライン番組表からの情報取得などの実現により，ユーザが行う操作や参照できる情報が爆発的に増加し，ボタンやスイッチ，メニュー階層などの増加により操作性が低下するという問題があった．そこで我々は，音声インタフェースの導入により情報家電の操作性の改善を試みた．

テレビ操作として望ましいものに番組タイトルや出演者名などによる選局や検索があるが，これらはオンライン番組表に正式名で登録されている情報がそのまま使われるとは限らず，略称や通称が使われる場合が多い．略称を受理できるようにするには，既存の略称を音声認識辞書に追加登録するのに加えて，日々新しく作られる略称に対応する必要がある．

そこで本研究では，正式名で与えられた番組名や個人名の略称を自動的に生成するシステム（以降，略語生成システムと呼ぶ）を構築することにより，これに対処することを考えている．日本語には，ある表現から一部を省略あるいは変形することにより，別の表現に短縮する事例が数多く存在する（以降，簡略前の表現を「原型」，簡略後の表現を「略語」と呼ぶ）．原型から略語が生成される際，いくつかの規則（以降,「生成規則」と呼ぶ）に則って生成されると考えられるが，どのような場合にどの生成規則が適用されるのかは分からない．そこで，各原型に適用されると考えられる生成規則を適用し，複数の略語の候補（以降,「略語候補」と呼ぶ）を生成した後，その中から略語らしさの高いものを推定することを考える．

略語に関する関連研究として，日本語の名詞間の類似度によりコーパスから略語と原型の対応関係を取得する手法[2]，コーパスからテンプレートを用いて抽出する手法[3]，略語生成の確率モデルを用いて抽出する手法[4]などが提案されてきた．[2]は，原型から略語への変換が規則に基づいて行われているため，規則を適用できない原型と略語は抽出できない．[3]は,「A を略して B」「A（以下，B と略す）」のように略語と原型の対応関係が明確に記述された文章からでないと抽出できない．[4]は，略語の生成精度が学習データの質に依存するため，高い生成精度を得るために大規模なデータ収集が必要となり，労力が掛かる．

2. 日本語の略語の生成規則

略語が生成される目的は発声時の労力を抑えることと所要時間の短縮である．使用頻度の高い語を毎回繰り返し言うのは面倒であるから，適当な長さに縮めて言い易くしようとする．ただし，極端に縮めてしまうと略語から原型を連想しにくいため，原型を連想できる最低限の長さを確保する必要がある．また，他の語あるいは他の略語と表面的に同音になることは避けるようにする．日本語の場合，大半の略語の長さは 2〜4 モーラに収まることが知られている[5]．

原型から数モーラを取り出す際にも規則性がある．原型が単一形態素語の場合は語頭から数モーラを略語とし，複合語（2 つ以上の形態素から構成される語）の場合はいくつかの形態素からそれらの先頭の数モーラずつを取り出して略語とする．どちらの場合も，語頭あるいは各形態素の先頭から数モーラを取り出す場合が圧倒的に多い．つまり，語頭あるいは要素の先頭から数モーラを取り出すことにより，原型との連想関係をできるだけ保ったまま表記を短縮することができる．

手作業で収集した原型-略語対に対し，どの生成規則でその略語ができているか調査した．図 1 は原型を構成する形態素が 1 つの場合の 63 語，図 2 は形態素が 2 つの場合の 374 語，図 3 は形態素が 3 つの場合の 109 語，図 4 は形態素が 4 つの場合の 18 語について調べた結果である．これらより，単一形態素語の場合は原型の先頭数モーラを取り出し，複合語の場合は先頭 2 モーラずつを取り出せば，大半の略語は生成できてしまうことが分かる．

以上の考察から，本研究では略語の生成規則を次のように考える．

原型が単一形態素語の場合，以下の生成規則により略語を生成できる場合がある．

【頭 n】　原型の先頭から n モーラを取り出す．

【尾 n】　原型の末尾から n モーラを取り出す．

2 形態素語の場合，更に以下の生成規則が加わる．

【1 形態素】　各形態素をそのまま略語とする．

【頭 m+ 頭 n】　2 つの形態素を選び，前から先頭 m モーラ，後から先頭 n モーラを取り出して繋げる．

【頭 m+ 尾 n】　2 つの形態素を選び，前から先頭 m モーラ，後から末尾 n モーラを取り出して繋げる．

【尾 m+ 頭 n】　2 つの形態素を選び，前から末尾 m モーラ，後から先頭 n モーラを取り出して繋げる．

【尾 m+ 尾 n】　2 つの形態素を選び，前から末尾 m モーラ，後から末尾 n モーラを取り出して繋げる．

3 形態素語の場合，更に以下の生成規則が加わる．

【形態素 + 形態素】　2 つの形態素を繋げる．

ただし，m, n の値はそれぞれ 2, 3, 4 とする．図 1 から図 4 までのすべての原型に対し，形態素の数に応じて生成規則を適用すると，略語の約 70%以上をカバーできる．

3. 生成規則による略語の自動生成

原型から略語が生成される際，大半の原型に対し前述の生成規則のどれかが対応するが，どのような原型に対しどの生成規則が適用されるかは分からない．このため，形態素解析により原型が何個の形態素から構成されているかを調べ，各場合に応じて適用できるすべての生成規則を適用する．単に略語のカバレッジを上げることが目的であれば，生成された略語候補の中に真の略語があれば良い．しかし，原型を構成

(注1)：本システムでは，音声入力により電源やチャンネルの切替，音量の調節，番組タイトルや出演者名によるオンライン番組表の情報検索などを行える．

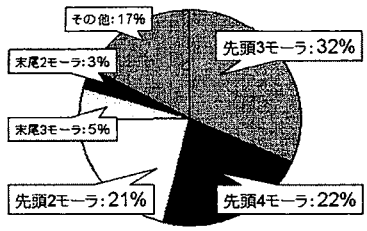

図1　単一形態素語に適用される生成規則

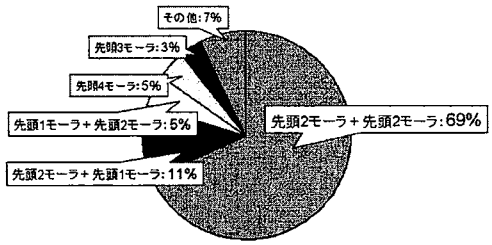

図2　2形態素語に適用される生成規則

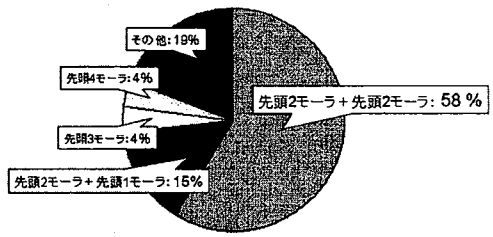

図3　3形態素語に適用される生成規則

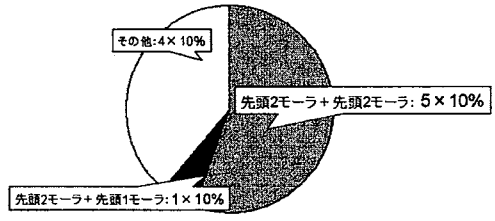

図4　4つ以上の形態素から成る複合語に適用される生成規則

する形態素が多くなればなるほど略語候補の数は爆発的に増加するため，認識辞書に略語候補を登録する際，既に登録されている語と衝突する，あるいは近くなる場合がある．そういう場合に，誤認識を避けるためにその略語候補を登録しないとすると，もしその略語候補が真の略語であった場合は，真の略語を認識辞書に登録できない危険が増える．そこで，生成された略語候補に対し以下のスコアリングを考える．

- 予備実験により，略語の各生成規則の適用されやすさは異なることが分かっている．そこで，適用されやすい生成規則により生成された略語候補には高いスコアを割り当てる．
- 新しく生成された略語のモーラ列の言語的特徴は，既存の略語のそれと類似していると考えられる．そこで，既存の略語のモーラ列による言語モデル（以降，「略語モデル」と呼ぶ）を作成し，各略語候補に対し略語モデルに基づいたスコアを計算する．
- 一般に使用頻度の高い略語候補ほど略語らしさが高いと考えられる．そこで，Web 検索(注2)により各略語候補の使用頻度を求め，その高さに応じてスコアを割り当てる．

一般に，文字で Web 検索を行う際の検索語の表記は，音を決めたとしても複数通り存在し得る（たとえば，「キムタク」，「キム拓」など）．このため，発音によって規定された語の Web 上での使用頻度を求める場合，考えられるすべての表記を用いて Web 検索を行う必要がある．ある語の使用頻度は，その各表記の使用頻度を加えたものであると考えられる．

略語候補を Web 検索の結果に基づいてスコアリングする際，各略語候補の実際の使用頻度を求めるには各略語候補の表記を網羅的に求めておくことが望ましい．なぜなら，一般に使われている表記が分からない場合，その略語候補の実際の使用頻度を求めることができないからである(注3)．しかし，提案法において原型に生成規則を適用すると，各形態素からモーラ列を取り出す際に漢字はモーラに分解され，元の漢字の情報は削ぎ落とされてしまう．そこで，各略語候補の発音が決まった段階で，部分的にでも漢字に戻せる部分があれば，それらをすべて挙げておきたい．しかし，形態素解析では漢字1字毎の読みを取得できない場合があり，その場合は何らかの方法により漢字1字毎の読みを求める必要がある．そこで，漢字とその読みを登録した漢字辞書(注4)を参照することにより，漢字1文字毎の読みを推定する．漢字辞書に登録されていない漢字が出現した場合，周囲の文字の読み及びその推定結果を用いて読みを推定する．また，形態素解析により形態素毎の読みは分かっているので，その結果を用いることにより読み推定の精度は高まると考えられる．

以上により，本研究では次節以降で述べる手順で略語を生成することにより，略語のカバレッジを上げる．

3.1　形態素への分解

一般に各原型に適用される生成規則は，原型が何個の形態素から構成されるのかにより異なる．そこで，既存の形態素解析システム(注5)により原型を形態素に分解し，同時に各形態素の表記，読み，品詞などを取得する．その際，原型から略語生成に不要と考えられる助詞，助動詞，記号を削除する．図5はその例である．

4.　略語候補の絞り込み

生成された略語候補に対し，生成規則の適用されやすさによるスコア，略語の言語モデルによるスコア，Web 上での使用頻度によるスコアを求め，それらにより各略語候補が真の

(注2)：本研究では，Web 検索に「Yahoo! JAPAN [8]」を用いた．
(注3)：たとえば木村拓哉の略語「キムタク」を検索すると，平仮名表記（きむたく）では 27,200 件，片仮名表記（キムタク）では 3,650,000 件，漢字と平仮名の表記（木む拓）では 4 件，漢字と片仮名の表記（木ム拓）では 13 件となる．
(注4)：本研究では，漢字辞書に [9] を用いており，約 2000 字の漢字の読みが登録されている．
(注5)：本研究では，形態素解析に「茶筌 [6]」を用いた．

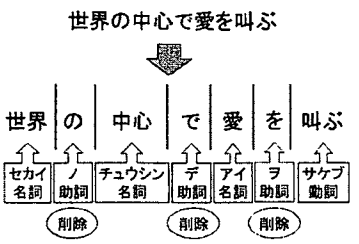

図 5　形態素への分解の例

表 1　各生成規則と重みの対応（単一形態素語）

生成規則	重み
頭 3	0.4
頭 4	0.3
頭 2	0.2
尾 3	0.05
その他	0.05

表 2　各生成規則と重みの対応（3 形態素語）

生成規則	重み
頭 2+ 頭 2	0.6
頭 2+ 頭 1	0.15
頭 1+ 頭 2	0.1
頭 4	0.1
その他	0.05

表 3　各生成規則と重みの対応（3 つの形態素）

生成規則	重み
頭 2+ 頭 2	0.5
頭 2+ 頭 1	0.25
頭 3	0.1
頭 4	0.1
その他	0.05

表 4　各生成規則と重みの対応（4 つ以上の形態素から成る複合語）

生成規則	重み
頭 2+ 頭 2	0.5
頭 2+ 頭 1	0.1
その他	0.4

略語である確率を求める．

4.1　生成規則による絞り込み

原型を構成する形態素の数と原型に適用された生成規則に応じ，各略語候補に重みを割り当てる．表 1 は原型が単一形態素語の場合，表 2 は 2 形態素語の場合，表 3 は 3 形態素語の場合，表 4 は 4 つ以上の形態素から成る複合語の場合について，それぞれ各形態素と重みの対応を示したものである．

4.2　略語モデルによる絞り込み

本研究では，インターネットから取得した略語 718 語を用いてトライグラムモデルを作成した．確率の平滑化にはバックオフ平滑化法を用い，ディスカウント係数の設定にはウィッテン・ベル法を用いた．これにより，各略語候補に対し略語モデルによるスコアを計算できる．

4.3　Web 検索による絞り込み

あるページ Y の中で文字列 X が出現する回数を $M(Y,X)$ と定義する．ある原型 B から生成された略語候補 C_i について，Web 上から原型 B を含むページ群 P_j を取り出す．このとき，各 P_j において $M(P_j,B) < M(P_j,C_i)$ であれば，略語候補 C_i は原型 B の真の略語であると見なす．

5.　略語のカバレッジと音声認識性能の評価

本研究が提案した略語自動生成法では，原型に適用させる生成規則を増やすと略語候補が爆発的に増加する場合がある．これにより真の略語を生成しやすくなる反面，略語生成機能を音声インターフェースに応用する場合，略語候補の増加が音声認識辞書のサイズ増大に繋がり，音声認識性能の低下や他の認識語彙との衝突といった問題が生じる危険がある．そこで，略語候補の数の増加が，真の略語の推定性能や音声認識性能に及ぼす影響について評価する．

略語生成システムにおいて，複数の生成規則により生成された略語候補の中に真の略語が含まれていれば，その略語はカバーできたと考える．そこで，略語の生成性能として

$$\text{カバレッジ} = \frac{\text{真の略語を生成できた原型の総数}}{\text{原型の総数}}$$

を考える．

音声認識システムにおいて，複数の音素列に対し同じ動作を割り当てている場合，入力音声がユーザの発話した音素列の通りに認識されなかったとしても，動作そのものはユーザの想定した通りになる場合がある．そこで，音声認識システムの動作がユーザが想定した通りであれば良いとする場合と，ユーザが発話した音素列の通りに認識されれば良いとする場合を考える．つまり，音声認識性能として

$$\text{想定動作率} = \frac{\text{システムがユーザの想定通りに動作した回数}}{\text{システムが動作した回数}}$$

$$\text{音声認識率} = \frac{\text{システムが入力音声を正しく認識した回数}}{\text{システムが入力音声を認識した回数}}$$

の 2 通りを考える．

5.1　評価条件

インターネットから原型-略語の対を 40 対取得し，提案法により原型から略語候補を生成した．ただし，形態素解析では茶筌 [6] の結果を手動で修正した．漢字の読みの推定では約 2,000 字の漢字と読みが登録された漢字辞書 [9] を用いた．略語候補の絞り込みでは，生成規則，略語モデル，Web 検索による尤度を用いた．

音声認識性能の評価で用いる音声データとして，男性 5 名と女性 6 名の計 11 名に対し，防音室内で接話マイクを使用して収録した．サンプリングレートは 16ksamples/sec である．発話内容は上記の真の略語 40 語である．雑音は加算しない．音声データの認識率は全話者とも 100%であり，その際の認識デコーダは Julian ver3.5.3 [10]，認識語彙数は 2,600 語である．音響モデルは JNAS の新聞記事読み上げ音声コーパスを用いて作成しており，学習に用いた音声データは男女ともに 150 名，発話数は 100 語である．学習は Hidden Marcov

Model Toolkit により行い，音響分析条件は 16kHz サンプリング，16bit 量子化，フレーム長 512 点，窓長 400 点，フレームシフト長 160 点，特徴パラメータは 25 次元 mfcc (12mfccs + pow + 12Δmfccs) である．

5.2 評価結果

各原型から生成された略語候補のうち，スコアの上位 N 個を認識辞書に登録したときの，略語のカバレッジと音声認識性能を調べた．各原型について上位 N 個の略語候補を採用したときのカバレッジ，想定動作率，音声認識率，及び認識辞書の登録語彙数を図??に示す．

採用する略語候補の数を増やすと，カバレッジ，想定動作率，音声認識率も上がるが，上がり方は徐々に緩慢になる．一方，認識辞書の登録語彙数が増えることにより，音声認識の処理にかかる負荷は急激に増えていく．採用する略語候補が上位 10 個程度のときが，略語生成性能と音声認識性能の妥協点になると考えられる．

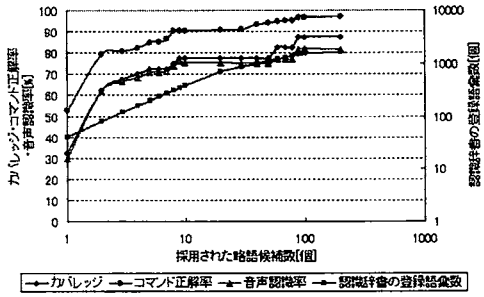

図 6 評価結果

6. 今後の課題

6.1 漢字の読みが変わる場合の略語生成

漢字から数モーラを取り出す際，原型と略語で読みが変わってしまう場合があるため，略語候補の表記は真の略語と同じでも読みが間違ってしまい，略語生成に失敗する場合がある．

6.2 Web 検索時の問題

原型と略語候補が同時に出現するページを検索する際，その略語候補を含み，かつそれより長い候補が他にある場合は，実際の使用頻度を求められない場合がある．

また，Web 検索の回数が多くなると，処理全体にかかる時間は爆発的に長くなる．スコアリングに Web 検索を用いる場合，Web 検索の回数を少なくする工夫が必要である．

謝辞

本研究の一部は同志社大学学術フロンティア研究開発プロジェクト「人間と生物の賢さの解明とその応用」の援助を受けた．

文献

[1] 榎 将功, 大田 健紘, 伊田 政樹, 長岡 宏, 松田 繁樹, 中村 哲, 木田 豊, 柳田 益造, "次世代情報家電への音声インタフェースの導入に関する検討",
[2] 酒井 浩之, 増山 繁, "略語とその原型語との対応関係のコーパスからの自動獲得手法の改良", 自然言語処理, Vol.12, No.4, pp.207-231, 2005.
[3] 桜井 裕, 佐藤 理史, "ワールドワイドウェブを利用した用語説明の自動生成", 情報処理学会論文誌, Vol.43, No.5, pp.1470-1480, 2002.
[4] 村山 紀史, 奥村 学, "Noisy-channel model を用いた略語自動推定", 言語処理学会, pp.763-766, 2006.
[5] 窪薗 晴夫, "もっと知りたい！ 日本語 新語はこうして作られる", 岩波書店, 2002.
[6] 茶筌, http:chasen.naist.jphikiChaSen
[7] JNAS, http:www.mibel.cs.tsukuba.ac.jpjnas
[8] Yahoo 検索 (ウェブ検索), http:developer.yahoo.co.jpquery_parameterswebsearch.html
[9] 玉岡 賢治雄, http:home.hiroshimau.ac.jpktamaokadown.htm
[10] Julian, http:julius.sourceforge.jp

付録

A 漢字 1 文字毎の読みの推定

原型が漢字を含む場合，漢字 1 文字毎の読みの推定は以下の手順で行われる．例を図 7 に示す．

（1） 原型の形態素解析により分解し各形態素について，表記の文字列を $C = c_1 c_2 \cdots c_n$ とする．日本語を対象とし，各 $c_i (1 \leq i \leq n)$ は漢字，平仮名，片仮名のいずれかであり，数字やアルファベットは含まないと仮定する．

（2） 各 c_i に対し正規表現 p_i を生成する．c_i が平仮名及び片仮名であれば読みは自明であるから(注6)，その読み K を用いて $p_i = K$ とする．c_i が漢字であれば漢字辞書（漢字とその読みが登録された辞書）を参照し，漢字の読みが登録されていれば，それらすべての読み $R_1 R_2 \cdots R_m$ を用いて $p_i = (R_1|R_2|\cdots|R_m|.+)$ とする．ただし，末尾の".+"は任意の文字列(注7)を表す．漢字辞書に登録されていなければ $p_i = .+$ とする．

（3） 形態素の読みと各 p_i を繋げた正規表現 $P = p_1 p_2 \cdots p_n$ のマッチングを行う(注8)．c_i が漢字であり，かつ p_i に合致した文字列 S_i が漢字辞書の読み $R_j (1 \leq j \leq m)$ と合致する場合，R_j を漢字 c_i の読みとする．S_i がどの R_j とも合致しない場合，すなわち S_i が任意の文字列として合致した場合は何もしない．

（4） これまでの処理で読みが確定した文字 c_i とその読みをそれぞれ区切りとし，原型の表記と読みを分解する．これにより，読みが確定していない文字列（漢字列）とその読み

(注6)：厳密には読みはすべて片仮名で表しており，原型に平仮名が含まれている場合は片仮名に変換する必要がある．読みを片仮名表記で統一している理由は，原型に「ヴ」「ヵ」「ヶ」が含まれている場合に対応するためである．
(注7)：プログラミング言語によっては，"."が任意の 1 バイト文字列を表し，全角片仮名は 2 バイトで表される場合がある．この場合は，任意の全角片仮名列に合致するための正規表現"(..)+"を使用すべきである．
(注8)：形態素毎にマッチングを行う理由は，正規表現が長くなればなるほどマッチングに失敗する危険が増えるためである．正規表現を形態素毎に区切ってマッチングすることにより，マッチングの成功率を上げている．

の対を得られる．各対について，漢字１文字毎の読みのモーラ数が等しくなるように，漢字に読みを割り当てる(注9)．

（5） 残りの形態素について最初から処理を繰り返す．

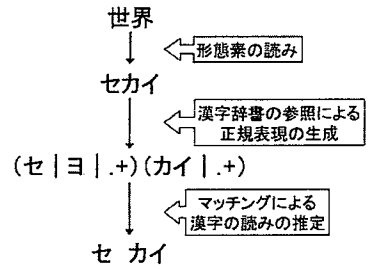

図７ 漢字１文字毎の読みの推定

(注9)：この処理は，原型に含まれる漢字が漢字辞書に登録されていない場合や，登録されていてもその漢字の読みが漢字辞書の読み候補と合致しない場合の救済措置である．