自然言語処理 152－22
（2002．11．13）

ユーザ文書を用いた個別かな漢字変換支援

田中久美子† 早川大地‡ 武市正人⊙ 玉井哲雄†
†東京大学 大学院 情報学環 ‡東京大学 大学院 学際情報学府 ⊙東京大学 大学院情報理工学系研究科
{kumiko,takeichi}@ipl.t.u-tokyo.ac.jp daichi@oak.dti.ne.jp
tamai@graco.c.u-tokyo.ac.jp

予測型入力システムは携帯電話の普及と共に、世界的にその重要度を増している。これら予測型入力においては、ユーザ固有の文章を予測型入力で扱う個別化技術に未だ研究の余地があり、方言や法律文書といった特徴のある文書の入力は難しい。本稿では、少量のユーザ文書をかな漢字変換システムに持たせることにより、ユーザ固有の文書入力を支援する機構を提案し、有効性と共に報告する。

Personalization of Predictive Entry Systems using a Small Amount of User Corpus

Kumiko TANAKA-Ishii Daichi Hayakawa Masato Takeichi Tetsuo Tamai
Graduate School of the University of Tokyo

As mobile phones become popular,predictive text entry methods are now an important technology world wide. Such methods still need research in the view of personalization,such that how to make the text entry software propose the vocabulary preferred by the user.
In this paper,we report our method for the personalization by using the small amount of unanalyzed text of the user. With this method, text of a dialect or of a certain domain can be entered more easily.
Having a user entry,the system dynamically cuts out the corresponding chunks from the user text and propose them among words proposed by the dictionary. We show our preliminary system and report its efficiency.

1 はじめに

文書の入力の方法には音声認識や文字認識などを利用したさまざまな可能性が広がっているが、現在では、未だキーボードによる入力が大半を占める。中国語や日本語では、総文字数がキーの数より遥かに多いため、言語の曖昧性を利用した予測型の入力方法が80年代に用いられるようになった。予測型の入力方法では、ユーザは曖昧な時系列、日本語ならかな列を入力し、システムがその時系列が何にあたるのかを辞書などのデータから統計的に予測して提示し、それを見てユーザが結果を確定する仕組みを用いる。

近年では、機器類の小型化がすすみ、特に携帯電話の普及を受け、世界的にキーの数が入力したい文字より少ないという現象下にある。その結果、予測型の入力は世界的な広がりを見せ、複数の北米の企業[1][9]が事業を展開するほか、研究面でもその有効性が問われている[3][6][7]。

予測型の入力における問題の一つに、個別化がある。すなわち、方言、古文、法律文書といった、特徴のある文書をいかに入力するか、という問題である。文章は、つまるところ個別のユーザが特定の文脈で入力するものである。したがって、これを支える入力システムも、ユーザに依存した語彙、文脈を扱う必要がある。たとえば、

ユーザに依存して、同じ入力に対する候補の順位は異なるべきであろうし、またユーザごとに語彙も異なる。これらをどのように予測型入力にとりいれていくかについては、未だ研究の余地がある。

現状での解決方法としては、候補の順序については、ユーザが直前に選んだ候補を最上位順位としたり、また、予測型入力システムの辞書に未登録の語 (未知語) については文字単位で何とか入力を行った上で、その語をユーザ辞書に登録することにより対処している。しかし、いずれも予測型入力ソフトウエアを使い始めてから使い勝手を良くするまでのユーザへの負荷は大きい。さらに、複数の異なる文脈を扱いたい場合には、複数のユーザ辞書を管理することなども求められる。

この問題を解決するために、予測型入力のための汎用の自動個別化技術を提案する。具体的には、予測型入力システムに少量のユーザ文書をもたせ、そこからユーザ入力に合致する表現を動的に抽出し、これまでの辞書に登録されている単語に加えて候補として提示するのである。これにより、未知語を減らす効果のほか、ユーザの文体を予測型入力システムに反映させることができる。

本稿では、まず次節においてユーザの文章編集のあり方に焦点をあてる。これを根拠として、上節で述べたシステム、およびそれを支える技術的な要について述べる。最後に本機構がどの程度有効であるのかの評価を行う。

2 ユーザの文書編集の特性

文章は一つの時系列となっている。時系列中のある語彙に焦点をあてると、それはその文章中のそれ以前に出現した語彙 (既出語)、あるいは、それ以前に出現していない語彙 (未出語) のいずれかとなる。当然のことながら、文章の頭の部分では、ほぼ全語彙が未出現の語彙で、文章の終わりに近づくにつれ既出語の割合は増える。

既出語の変化を考察するため、文章の先頭から一定の地点の既出語の割合を調べる。本論文では個別化に焦点をあてるため、特徴のある文章を 3 種類用意した。

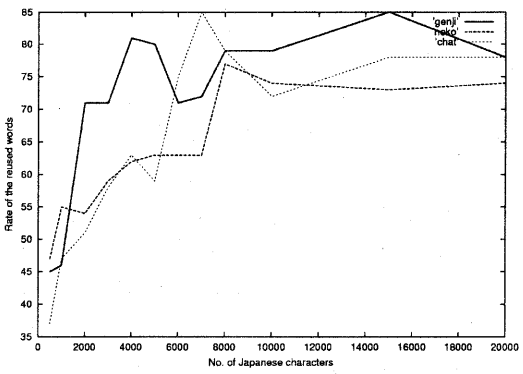

図 1: 既出単語の変化の近似

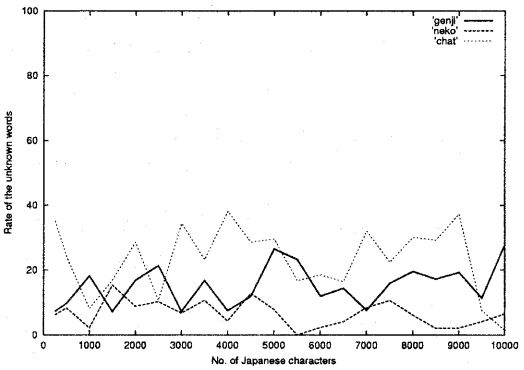

図 2: 未知語割合分布の近似

- 我輩は猫である
- 源氏物語
- チャット

これらの文章それぞれにつき、文頭からの位置と既出語の割合を近似的に示したものが図 1 である。横軸に文章の位置 (文字数) をとり、縦軸にその位置前後 1000 文字中の既出語の割合を調べ、平均値をプロットした (はじめから 1500 文字のところのプロットは、1000 文字から 2000 文字目までのプロットとなる)。上の文章は形態素解析 [11] で必ずしも解析可能ではないが、形態素解析器が提案するものを当座単語とみなして処理したため、近似的なグラフである。

予想どおり文章の最初の 8000 文字程度においては、未出語が多い。それを超えると、7 割以上が既出となる。つまり、文章の編集とは、文頭を除くと、既出の語彙 7 割と、未出の単語 3 割

を用いて行われるということになる。この傾向は、3つのまったく異なる文章によって多少の差はあるものの、極めて類似している。このように、既出語の割合が文章の頭に近い部分で7割に収束するのであれば、1万文字程度のユーザ文書を用いて予測型入力システムの精度向上を期待することができる。

次に未知語について考察する。未知語には未出語のものと既出語のものがある。このうち未知語で既出のものについては、ユーザ文書を利用することにより入力を効率的に行うことができる可能性がある。この準備として、文章中の未知語の分布をここでは調べることにする。図2は、文頭からの位置（文字数）を横軸、縦軸に未知語の割合を示したものである。尚、ここでは未知語を形態素解析の辞書に含まれない語とした。

実験に用いた文章は形態素解析 [11] で必ずしも解析可能ではないため、未知語は以下のように近似的に推定した。まず1500文字程度の文章を人手で解析し、未知語数 n を算出する。一方で形態素解析を行い、解析器の示す未知語と報告されるものの数 m を調べる。そして、その比 $\gamma = \frac{n}{m}$ を算出する。これを異なる1500文字の文章5つにつき行って γ の平均値を求める。その上で、文頭から x 文字の位置の前後1000文字の文章の形態素解析を行い、未知語として報告された数 m' に対して、全未知語数を $n = m' \times \gamma$ により推定する。(x, n) をプロットしたのが本図となる。

これを見ると、未知語の数は文章によって相当異なる。古文である源氏物語では未知語割合は極めて大きい。しかし、いずれの場合にも、未知語の分布は文頭だから多いということはなく、文章の位置に限らず現れる。ということは、未知語についても語彙の7割を再利用して編集されるとする仮説を適用するならば、少量のユーザ文書により、かなりの割合の未知語を捉えられる可能性は大きい。

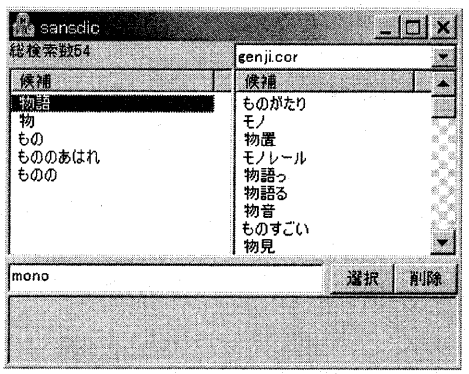

図 3: 源氏物語:「もののあはれ」を入力

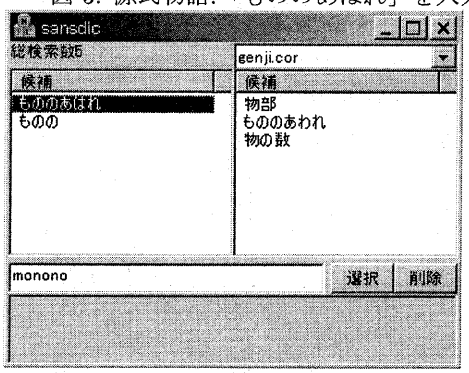

図 4: 源氏物語:「もののあはれ」を入力

3 個別支援入力の概要

前節をふまえて作成したシステム *sansdic* を示す。これは、将来はかな漢字変換のフロントエンドシステムとして動かすことを期待しているものである。現状では、その基礎的な性能を調べるため、ローマ字入力での補完を用いた単語単位の変換に対して実装したが、将来的には通常の連文節変換のフロントエンドとしても実装が可能である。

システムは、かな漢字変換用の単語辞書、SKKの文字辞書を読み込んで初期化される。システムが立ち上がると、ユーザはまずこれから入力を行う文章と同種の文章を設定メニューを開いてセットする。ユーザ文書は単なるテキストデータであれば何でもよく、未解析のままでよい。現状では図3では、源氏物語を5000文字分セットしてある。

ユーザは文書入力を行う。ここでは、「もののあはれを源氏...」を入力してみる。まず、最初は「もの」であるので、「mono」をユーザは図の下

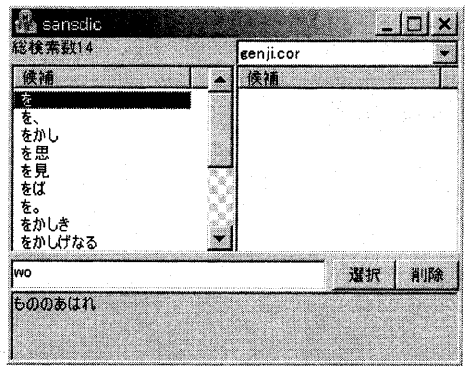

図 5: 源氏物語:「を」を入力

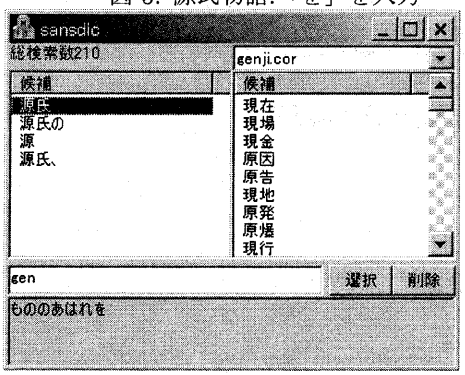

図 6: 源氏物語:「源氏」を入力

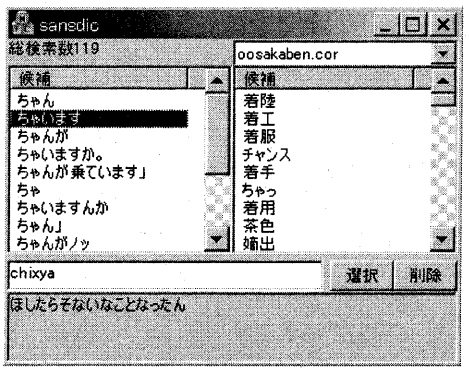

図 7: 大阪弁:「ちゃいます」を入力

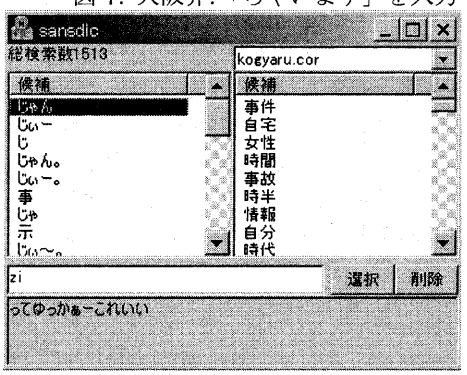

図 8: チャット:「じゃん」を入力

方の横長の部分に入力する (図 3)。すると、システムは文字辞書を用いて「mono」から始まる文字列「物語」「物」「者」等の語をユーザ文書から得て、これらが左の候補窓に示される。同時に右の窓には、辞書から得た mono で始まる単語が示されている。ここでは補完を用いているため、「mono」よりも長い単語が候補として挙がる。「もののあはれ」は 4 番目に出現しているので、この時点で候補を選択しても良いし、あるいはさらに続けて語「no」を入力して候補を絞りこむこともできる。絞り込む場合には、「もののあはれ」が最上位の候補として表示される (図 4)。

上下キー、および左右の箱間はタブキーで移動することにより、ユーザは自分の望む候補をハイライトさせる。この場合には、「もののあはれ」が、ユーザ文章候補群の 1 位に上がっているので、これを選択する。選択された文書は、最下段の横長の箱の中に現れる。

このように、ユーザはそれまでの文章を再利用して入力を行う。「もののあはれ」は、辞書にはない未知語であり、これが入力できたことになる。さらに、また「もののあはれ」は複数単語から成る語であるが、これが 1 単語ないしは 2 単語の入力で一度に変換可能であるということもわかる。

つぎに「wo」を入力する (図 5)。ここでは、候補の最上位に目的の単語が得られているので、これを選択する。図 6 では「源氏」という単語が「gen」だけで入力された様子が示されている。

同様に図 7 では大阪弁での入力を、図 8 ではチャットの入力の様子を示す。いずれもユーザ文書 6000 文字ずつをシステムにセットした上で入力を行っている。このように、ユーザ文書をセットしておくことにより、ユーザの選びそうな候補が上位に挙がり、また、ユーザ固有の未知語に対処できていることがわかる。

以上の動作から、本システムの要となるのは、つぎの二つである。

- 候補の切り出し
- 候補の提示順

次節ではこの点をシステムがいかに行っている

かについて説明する。

4 予測モデル

4.1 単語の切り出し

候補の切り出しと関連して、コーパスからのngram抽出がさかんにこれまで研究されてきた。しかし、本稿の主題は、大量のコーパスからの抽出ではなく、あくまで小さな文書数千〜数万単語の候補の切り出しである点が異なるといえる。

信憑性のある統計を得ることができないほどに小さな文書からの候補の切り出しには、まずは重複文字列が考えられる。重複文字列を単語相当文字列として捉える手法は、古くから文書圧縮の分野でさかんに行われてきた[2]。文書圧縮は言語に内在する予測可能な構造を捉えることにより実現されるから、本稿でも有用な参考手法となる。実際、自然言語処理の分野でも長尾ら[5]がこれに類する手法で重複文字列を用いて大量のコーパスからn-gramを抽出することを試みている。

以上の理由から、本稿でも重複文字列を候補とするが、といって、すべてを候補としたのでは、数が多くなってしまう。そこで、本稿では特に、つぎに定義する接頭最長重複文字列を候補とするものとした。接頭最長重複文字列とは、重複文字列のうち、他の候補の接頭辞をなさない文字列と定義する。たとえば「abracadabra」をユーザ文書とし、ユーザがaと入力したものとすると、aで始まる重複する文字列は、括弧の内を頻度として、

abra(2),abr(2),ab(2),a(5)

の4種類である。これらが候補となりえるが、このうちabrとab,および、aのうちの2回はabraの接頭部分であるため除き、結果、

abra(2),a(3)

が候補となる。aのうちの残りの2回もabraの一部ではあるが、これはabraの接頭をなすものではないため残す。

接頭最長重複文字列は、suffix array[4]と相性がよく、高速に文字列を切り出すことができる。以上より、システムの動作は以下のようなものとなる。ユーザが切り出したい候補の接頭部分を提示すると、システムはsuffix arrayを2分探索する。得られた候補を次節で説明する方法により並べ替え、ユーザに提示する。ユーザの入力が進むにつれて、ユーザ文書の大きさは動的に増える。その際、suffix arrayも動的に更新され、これらはすべて文書の最後に付加され新しい候補の提示に用いられる。

4.2 切り出した単語の並べ替え

切り出した候補は整列されて提示される。これは頻度を元に行われるが、この頻度は前述のように小さなコーパスから計算しなければならないという特徴がある。小さな文書でも有効な確率補正の方法として、過去にBellらによりPPMが提案されている[2]。PPMはもともと、圧縮の分野で提唱されたもので、わずか数語であっても確率を補正することができる点にその特徴がある。自然言語分野への応用でもかな漢字変換の候補をPPMを用いて補正すると、より良い候補が上位に挙がることが検証されている[10]。

以下、PPMを概説する。ある時系列$S=x_0 \ldots x_i$が与えられた時、つぎの要素x_{i+1}の確率を次式で推定する。

$$P(x_{i+1}) = \sum_{k=-1}^{kmax} u_k P_k(x_{i+1}) \qquad (1)$$

ただし、kはx_{i+1}の前k単語のことを表し[1]、特にk = -1の時は、文脈から情報が全く得られない場合に基本として参照される初期確率を表現する。そして、Bellらは、すべてのx_iについて等確率にする手法を一例としてあげている。$P_k(x_{i+1})$は前k単語を考慮した確率で、C_kをその文脈の現れた回数、$c_k(x_{i+1})$をその文脈でx_{i+1}が現れた回数とすれば、

$$P_k(x_{i+1}) = \frac{c_k(x_{i+1})}{C_k} \qquad (2)$$

と近似される。

ここで問題となるのは、ユーザ文書は未解析の文書を前提としており、単語に分かれていな

[1]通常自然言語処理で言われるn-gramは、PPMでは、k=$(n-1)$となる。たとえば、k=1の時は、自然言語処理におけるbigramを、k=2の時は、自然言語処理におけるtrigramを意味する。

いため、x_i をどのような単位で捉えるかという点である。本稿では、

- 文脈としてとらえる場合には x_i を文字としてとらえ、
- 候補としてとらえる場合には、x_i を切り出した文字列全体とみなす

ものとした。PPM を計算する際の候補の頻度は前節で述べた頻度を用いた。むろん x_i をすべて文字としてとらえる方法もあるが、候補は接頭部分を共有するため、文字単位では判別力が得られないため、このようにした。

u_k は直前の k 単語を足し込む重みである。重みの決め方にはさまざまな研究成果がある [8]。本稿の問題の場合には、PPMA[2] を用いるものとした。k の上限 $kmax$ は文献 [8] の成果をそのまま採用し 4(5-gram) とした。

5 評価

本節では、これまでに述べた方法による入力効率の向上に関する評価を行う。そのために、§2 に示した、3 種のデータにつき自動入力実験を行った。

まず、3 種のデータを文単位で切り、文をランダムに整列する。その上で、異なる大きさのユーザ文書を前から必要分だけ切り取って作った。このとき、大きいものは小さいものを含むようにユーザ文書を作成した。ユーザ文書は実際は偏っているものであるが、評価においては平均的な効率の向上を調べるため、文をランダムにソートした。

さらに、テスト文書として、およそ 1000 単語程度のものを学習コーパスとは異なる部分から切り出した。その文書を形態素解析 [11] を用いて解析し、解析誤りをすべて手動で直しローマ字読みを付した。以上の学習データとテストデータの組を、文書全体をランダムに並べ替えることを 5 回行って、5 組作り、入力実験を行う。

自動入力機能は、つぎの手順で行う。

1. まずその時点でに入力したい先頭の単語の読みをすべて入力する。(「もの/の/あはれ/を」を入力したい場合、「mono」を入力)
2. 候補の中から求める単語列で上位 5 位までに表示されたもののうち最長のものを選択する。(1 番目に「ものの」、2 番目に「もののあはれ」という候補が挙がった場合 2 番目の「もののあはれ」を選択する。5 番目までに選択肢が無く、かつ候補が 5 以上あった場合はさらに絞込検索を行う（この場合は「monono （ものの)」と入力する)。

ユーザ文書からの候補については、複数単語にわたって候補が表示される場合がある。たとえば「ものの」が一回で入力できた場合は、「あはれ」から入力を再開する。

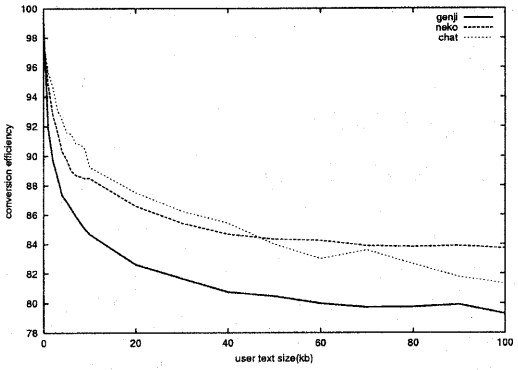

図 9: 変換効率

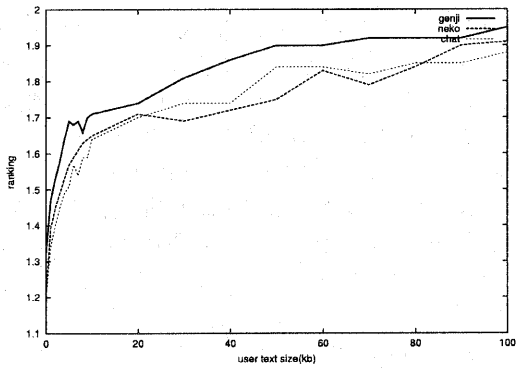

図 10: 候補の平均出現位置

5.1 変換効率と単語出現位置

本稿の提案を用いると、変換効率は 2 つの観点でよくなるはずである。第 1 に、変換回数の減少である。本稿では、もととしているかな漢字変換は単語単位の変換である。とすると、総変換回

数は、ユーザ文書からの候補提示がなければ単語の数とほぼ近くなる（ただし、助詞「に」「て」など、字面のみでは1語の「にて」とも「に＋て」の2語とも解釈できるこのような単語はまとまって検出されることがある。しかしその率はおおむね1%程度である。）。しかし、本稿での提案を用いると複数単語が一度に変換されることが増えるため、総変換回数は、用いない場合と比べて減ることになる。

ユーザ文書の大きさと変換回数の減少率の関係を調べた結果を図9に示す。ここでは、入力が不可能であった未知語を含めて総変換回数を計測する。横軸はユーザ文書の大きさ(単位KB)、縦軸は変換回数の減少率である(100単語を80回で入力すると、80%となる。)。

3つの結果が示されており、40KB程度まで、上から「チャット(chat)」「我輩は猫である(neko)」、「源氏物語(genji)」となっている（40kbを越したあたりで「チャット」と「我輩は猫である」は順序が入れ替わる）。左端の軸に接している点はユーザ文書を用いない場合の変換回数となっており、いずれも99%前後となる。変換回数が最大で80%程度となっているが、これはすなわち、1回の変換あたりおよそ(10/8=)1.25単語が変換された計算となる。10kbyteの文書は約2500単語に相当するが、たったそれだけのユーザ文書を用いることにより、この程度文節変換化されることを示している。

具体的に一回で変換されたものとしては、「御/ありさま/は」、「甘木/先生」、「本当/に/お/疲れ」などといったものがあった。すなわち、本手法により単語のつながりをも捕らえることができることがわかる。

候補を選ぶ際の評価指標は、総変換回数だけではなく、候補が何番目に現れるかという点をもふまえて評価する必要がある。つまり、総変換回数が増えても、1回の候補選択において、ユーザが10番目の候補を毎回選んでいるのでは、効果を得られるとは言い難い。これを確認するために、図10に、n番目の候補を選んだとしてnの平均値を示す。横軸はユーザ文書の大きさ、縦軸はnの平均値である。ユーザー文書を最大で100kbほどに取った場合平均1〜2番目の候補が選ばれている。しかし、nは収束しつつあるとはいえ、増加の傾向にある。増加する原因は、ユーザ文書量を増やすとやはり候補が増えてしまい、これらを整列するためのモデルが現状では不十分であるからである。この点は今後の重大な課題の一つである。

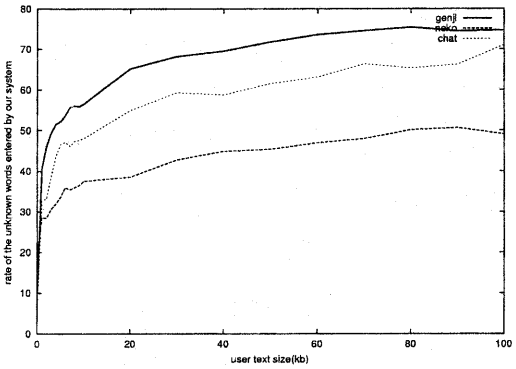

図 11: 入力可能な未知語の割合

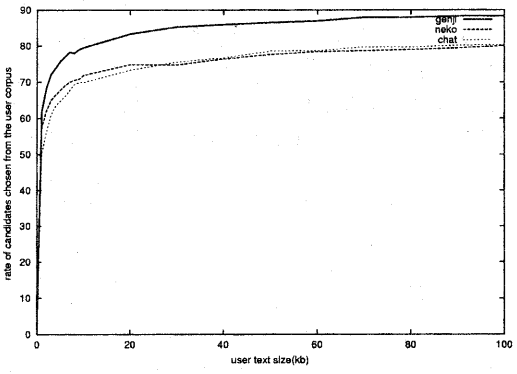

図 12: ユーザ文書からの候補選択率

5.2 未知語の入力可能性

図11は本手法により入力可能となった未知語の割合を示している。「我輩は猫である」は3つの文書のうちもっとも現代文に近く、発生する未知語も「癒る」などの旧字体や「気遣（キズカイ）」などの送り仮名の違いなどのケースに限られ、未知語自体が少ないため、結果はそれほど高くない。一方で、「源氏物語」「チャット」では、「たまへり」や「おつー（おつかれさまの意）」などの未知語が繰り返し登場し、入力可能な未知語はユーザ文書70Kバイト(1700〜1800

単語) 程度のところまで順調に増加し、最終的に100kbのコーパスを使用した場合には7割から8割ほど入力することが出来た。すなわち、語彙の7割が再利用で、未知語の発生は文書の位置と関係のない、という§2で得られた結果からも妥当な結論である。

5.3 ユーザ文書の利用率

§3で示した利用例では、ユーザ文書からの候補が選ばれる場合と、辞書から提示する候補が選ばれる場合がある。そこで、ユーザ文書のものが選ばれる割合を示したのが図12である。ユーザ文書を増やすと、利用率も増え、特に源氏のように文体が特徴的なものは、利用率が85%を超える。この結果はやはり§2を裏付ける結果となっている。

6 結論

予測型入力には、中国語や日本語の入力システムの他、近年の機器類の小型化を受けて携帯電話上などで使用される入力システムなど、世界的に広まりを見せている。この入力方式においては、未だ方言、古文、法律文書などといった特徴的な文章の入力が難しいという問題がある。本稿では、汎用の個別化手法を提案し、この問題を解決することを試みた。具体的には、少量のユーザ文書を用い、ユーザ文書から動的にユーザ入力に合致する可能性のある文字列を切り出して提示する。

予測型変換システムのフロントエンドとして動くシステム例を提示した後、かな漢字変換について自動入力実験を行い、その有効性を確認した。変換効率が向上し、未知語の発生が抑えられることが示された。

今後は候補の切り出しと整列の方法をさらに研究した上で、ほかの予測型変換入力に対する効果を確認すると共に、かな漢字変換システムのフロントエンドシステムとしての本格的なシステム作りを検討したい。

参考文献

[1] Tegic 9. Tegic 9 home page, 2000. http://www.t9.com.

[2] T.C. Bell, J.G. Cleary, and I. H. Witten. *Text Compression.* Prentice Hall, 1990.

[3] C. James and K. Reischel. Text input for mobile devices: Comparing model prediction to actual performance. In *Proceedings of the Annual SIGCHI Conference*, 2001.

[4] U. Manber and G. Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal of Computing*, 1993.

[5] M. Nagao and S. Mori. A new method of n-gram statistics for large number of n and automatic extraction of words and phrases from large text data of japanese. *COLING*, 1994.

[6] K. Tanaka-Ishii, Y. Inutsuka, and M. Takeichi. Japanese input system with digits–Can Japanese be input only with consonants?—. In *Human Language Technology Conference 2001*, 2000.

[7] K. Tanaka-Ishii, Y. Inutsuka, and M. Takeichi. Entering text with a four-button device. *International Conference on Computational Lingguistics*, pages 933–935, 2002.

[8] W.J. et al. Teahan. Probability estimation for ppm. In *NZCSRSC'95*, 2000. http://www.cs.waikato.ac.nz/ wjt/papers/NZCSRSC.ps.gz.

[9] ZI-Corp. Zi home page, 2000. Available from http://207.229.18.241/.

[10] 丸山卓久, 田中久美子, and 武市正人. Ppm法を用いたかな漢字変換の学習モデル. In **情報処理学会自然言語処理研究会論文集**, volume 第1 4 6回, 2001.

[11] 松本裕治 他. 日本語形態素解析システム「茶筅」使用説明書, 1997. 奈良先端科学技術大学院大学 Technical Report.