
日本語動的単語補完方式 **Nanashiki** を活用した予測入力

Predictive Text Input with Japanese Dynamic Abbreviation Expansion Method “Nanashiki”

小松 弘幸 高林 哲 増井 俊之*

Summary. Existing predictive text input systems usually uses a static word dictionary or the history of word usage for predicting the user’s next input word. In addition to the dictionary and the history data, it is also useful to predict the input word from other texts. For example, a user can efficiently compose a reply message, using the original message for predicting words used in the reply message.

Although this technique, called the *dynamic abbreviation expansion*, is popular for editing English texts, it was not applicable for editing Japanese texts, where Kana-Kanji conversion is commonly used for text input. We introduce a technique *Nanashiki* that enables dynamic abbreviation expansion for Japanese texts. With our technique, Japanese words which appear in a text can be used as the candidate words for predictive text input, and users can efficiently compose a reply message using the context information of the original message.

1 はじめに

POBox[1] に代表される予測入力方式は、かな漢字変換のパッケージソフトや携帯電話の入力方式として広く普及している。

従来の予測入力方式で予測の対象になる候補は、主に利用者の過去の入力に基づいて学習された単語である。しかし、過去の入力にのみ基づいた方式では、効果的な予測が行えない場面が存在する。例えば、メールの返信をする場面では、相手のメールに含まれる単語を用いることが多い。しかし利用者がその単語を過去に入力していないと、優先度の低い候補として提示されるか、まったくの未知語であるためそもそも候補として提示されない。

この問題の解決方法として、相手のメッセージを含めたコンテキストを捉える目的で、入力している文章の周辺および関連する文章内に存在する単語を候補に加える方法が考えられる。例えば、相手のメッセージに「予測入力方式」や「コンテキスト」という単語が含まれる場合、利用者が返信の文章の作成する時に、これらの単語を優先的に提示する。

* Hiroyuki Komatsu, 東京工業大学情報理工学研究所数理・計算科学専攻, komatsu@taiyaki.org, Satoru Takabayashi, ソニーコンピュータサイエンス研究所, 奈良先端科学技術大学院大学情報科学研究科, satoru@csl.sony.co.jp, Toshiyuki Masui, ソニーコンピュータサイエンス研究所, masui@csl.sony.co.jp

この方法は動的単語補完の応用といえ、テキストエディタ Emacs では dabbrev[3] と呼ばれて広く用いられている。しかし日本語での動的単語補完入力、かな漢字変換の作業と、文章からの単語の切り出しが必要になるため、実用的ではなかった。

本論文で述べる Nanashiki「七色」は日本語での動的単語補完入力方法を日本語インクリメンタル検索手法 Migemo[4] の応用により実現した手法である。POBox が提示する予測候補に Nanashiki により得られた単語を加えることにより、未知語もしくは低い優先度であった候補を効果的に利用者に提示することを可能とした。

2 コンテキストを捉えた予測入力

従来の予測入力方式が優先的に提示する候補は、利用者が過去に入力したことのある単語に限られている。例えば図1で示すように、メールに返信をする場合、相手のメッセージに含まれている単語は利用者が入力した単語ではないため、優先度の高い予測候補として提示されない。しかし、メールの返信というコンテキストを考えれば、相手のメッセージに含まれる単語は優先されるべきである。

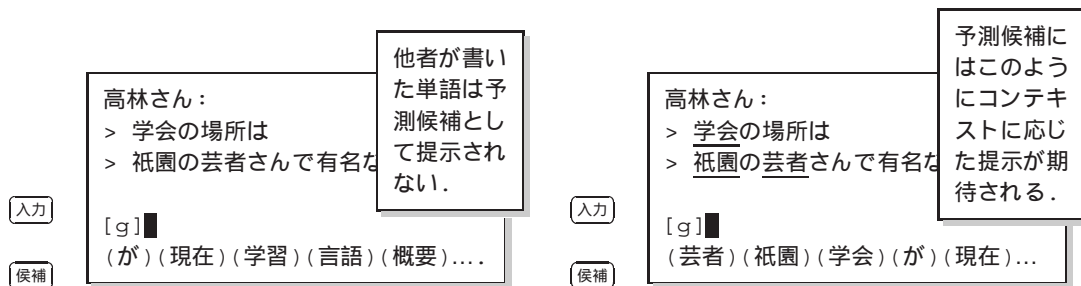


図1. 期待される予測候補の提示

利用者が一人で文章を作成している場面では、適切な予測候補の提示は問題なく行われる。なぜなら、文章を書くにつれて利用者が入力した単語が自然に学習されるからである。しかし、他者が書いた文章の引用を伴う場合、コンテキストに応じて適切に候補の提示を行うのは従来の方法では不可能である。

この問題を解決し適切な候補を提示するために、入力位置周辺の文章などから候補となりうる単語を抽出し、その単語を予測候補に加えるという方式を提案する。この方式は動的単語補完の技術を応用すると実現できる。しかし、日本語における動的単語補完は、かな漢字変換に由来する問題から、これまで実用的ではなかった。次節でその理由と今回解決した手法を述べる。

3 Nanashiki による日本語の動的単語補完

動的単語補完とは、過去に入力したことのある単語を少ない打鍵数で入力するための機構である。例えば、“abbreviation”を再び入力する際に [a] [b] [b] [EXPAND] と打鍵すれば “abb” が “abbreviation” に展開される¹。また動的単語補完は打鍵数を減らすだけでなく、打ち間違いを減らすという点でも有効である。

¹ [EXPAND] は動的単語補完の実行の指示を表す

3.1 日本語入力における動的単語補完の問題点

日本語入力における動的単語補完を実現しようとする場合、通常の英語入力における場合にはない、以下に示す問題が存在する。

- 1) かな漢字変換の手間
- 2) 補完候補となる単語の切り出し

第一の問題は日本語入力には、英語入力とは違い、「読みの入力 変換 確定」というかな漢字変換の手順が必要なために発生する。例えば「東京工業大学」という文字列を従来の動的単語補完によって入力する場合、「東京」と入力してから「東京工業大学」に展開するといった手順を踏む。しかし「東京」と入力するためには、まずその読みである「toukyou」を入力した後、読みを漢字へ変換しなければならない。すなわち、従来の動的単語補完は

1. 「toukyou」と読みを入力
2. 「東京」に変換、確定
3. 「東京工業大学」へと動的単語補完

という3段階の手順を踏む。しかし、この動的単語補完を用いた手順と、以下に示す通常の入力手順である

1. 「toukyoukougyoudaigaku」と読みを入力
2. 「東京工業大学」に変換

を比較しても、動的単語補完を用いた手順の入力効率は、かな漢字変換が煩雑なため、ほとんど向上しない。

そのため入力の効率化を目指すには、かな漢字変換を省いた

1. 「touky」と先頭の数文字を入力
2. 「東京工業大学」に展開

という動的単語補完が望ましい。しかし、そのためには「touky」という文字列から「東京工業大学」という文字列を探し出す仕組みが必要である。

加えて、日本語の動的単語補完における第二の問題として「補完候補となる単語の切り出し」がある。文章中から単語を切り出す処理は、英語の場合、空白という明確な単語間の区切りがあるため容易である。しかし、日本語の文章には明確な単語間の区切りがないため、文章を適切な単語群に切り分ける方法が必要がある。

これらの問題点を解決するために、かな漢字変換を省略した動的単語補完手法であるNanashikiを、日本語インクリメンタル検索手法Migemo²の応用、およびKAKASI³や茶筌⁴ [5]などのわかち書き用ソフトウェアを用いて実現した。

3.2 Migemo: 日本語のインクリメンタル検索

我々が開発したMigemoは、かな漢字変換を省略し、ローマ字のままの日本語のインクリメンタル検索を実現している。図2にMigemoを用いてキーワード「奇怪」をインクリメンタル検索する過程を示す。

² <http://migemo.namazu.org/>

³ <http://kakasi.namazu.org/>

⁴ <http://chasen.aist-nara.ac.jp/>

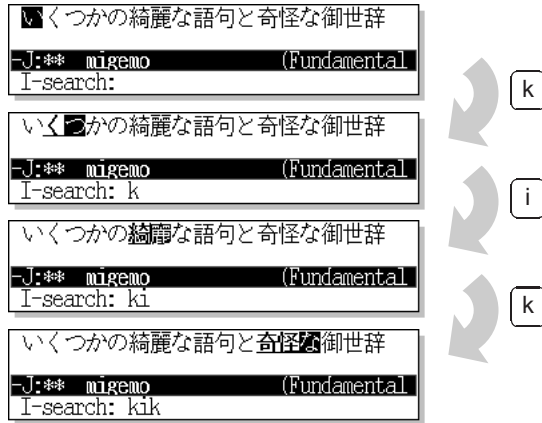


図 2. 「奇怪」を Migemo でインクリメンタル検索

この例では、利用者は **k****i****k** と入力し、かな漢字変換の作業を省いて「奇怪」の位置に到達している。一方、かな漢字変換を伴う検索では“kikai”とすべて入力した後に同音異義語の中から「奇怪」を選ぶという手順を踏む。このように Migemo では、かな漢字変換の作業に煩わされることなく、スムーズな日本語のインクリメンタル検索が可能である。

Migemo は利用者が 1 文字入力するごとに内部で動的に正規表現を展開し、その正規表現を用いて検索を行う。上の「奇怪」に対するインクリメンタル検索のための正規表現は図 3 のように展開される。最初の正規表現では“k”にマッチするテキストの位置へ、中央の正規表現では“ki”にマッチする位置へ、最後の正規表現では“kik”にマッチする位置へと、インクリメンタル検索が進む。

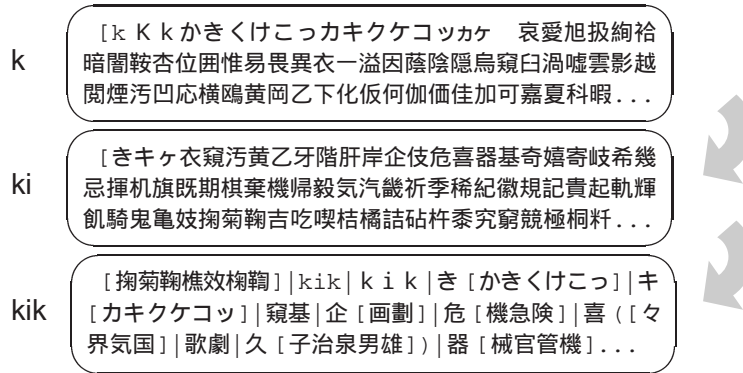


図 3. k, ki, kik に対する正規表現

3.3 Nanashiki

Nanashiki は動的単語補完での検索に Migemo の持つ正規表現展開の機構を用いることにより、かな漢字変換を省略した検索を実現した。正規表現展開により、第一の問題

点である「かな漢字変換の手間」は解決される。しかしまだ、「補完候補となる単語の切り出し」という第二の問題が残る。

正規表現展開により検索された文字列が、予測候補として適しているか不明である。予測候補として適した単語は、検索された文字列を単語の先頭に含んでいるものだけである。例えば、「京都府」と「東京都」の両単語は、どちらも「kyou」という読みで「京」が検索される。「京都府」は「京」を先頭に含んでいるため予測候補となるが「東京都」は「京」から始まる単語ではないため予測候補には適さない。

我々の手法では、検索された文字列の周囲の文章を単語ごとに区切ることによって、先頭の文字列かどうかを判定している。これら一連の動作の例を図4に示す。この例では、「今日の東京都の気温」という文章に続いて「k」を入力したため「今」、「京」、「気」が検索された。そして「今」と「気」を先頭の文字に含む、「今日」と「気温」が予測候補として追加される。「東京都」は「京」を含んでいるが、単語の途中の文字であるため無視される。

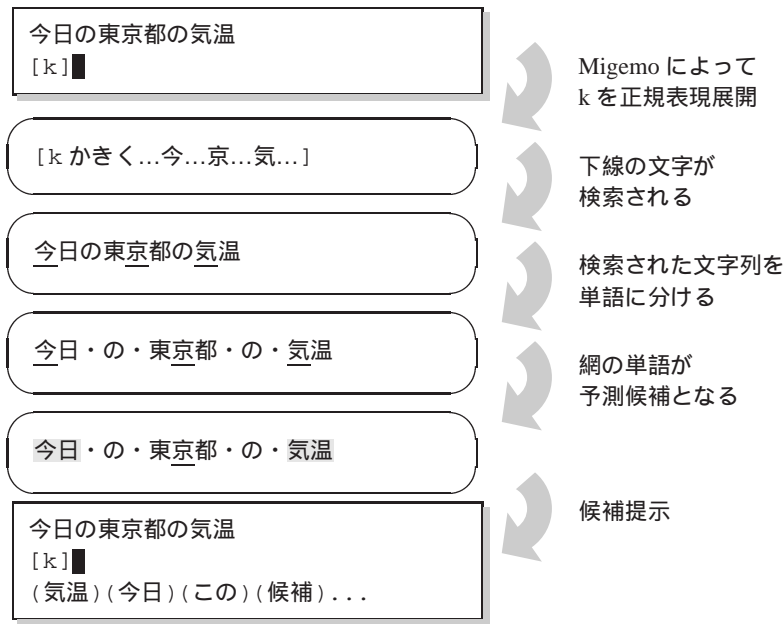


図4. Nanashiki による動的単語補完

4 実現

Nanashiki による動的単語補完を用いたテキスト入力の実現は、新たに作成した Emacs 用 POBox⁵ (図5) を用いて行った。Emacs 用 POBox は複数のかな漢字変換用機構をモジュールとして統合的に扱う機能を持つ。そのため、従来の POBox サーバモジュールに加えて、さまざまな変換用モジュールを組み込むことが可能である。動的単語補完による予測候補の追加の仕組みも、ひとつのかな漢字変換用モジュールとして実現した。

⁵ <http://www.taiyaki.org/pobox/>

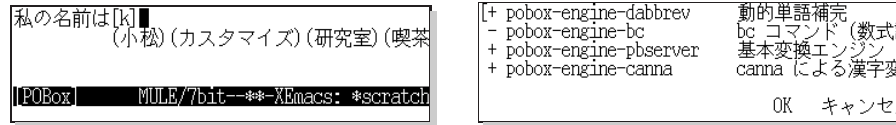


図 5. Emacs 用 POBox (左:動作画面, 右:複数のかな漢字変換機構の選択)

今回の実装では動的単語補完モジュールが予測対象としてコンテキストを検索する文書は, Emacs が現在編集集中の文書とした. 更に幅広い文書を対象とすることも可能であるが, 利用者にストレスを与えない処理時間を実現するために対象文書を限定した.

5 評価

コンテキストを考慮した動的単語補完による入力効率の向上を, メールのやりとりにおける相手のメッセージに含まれる単語の再利用率ととらえ, 次の手順で測定した.

メッセージ分割 メールを相手のメッセージ部分である引用部分と, 利用者が作成した文章部分に分割する.

共通単語の抽出 分割したそれぞれのメッセージから, 単語を抜き出し, 共通して含まれている単語を取得する.

再利用率を計算 返信メッセージに含まれる単語のうちの共通して含まれている単語の割合を, 動的単語補完によって提示される単語の割合として求める.

名詞がコンテキストを形成していると仮定し, 本評価では調査する単語を名詞に限定した⁶. 図 6 に評価の例を示す.

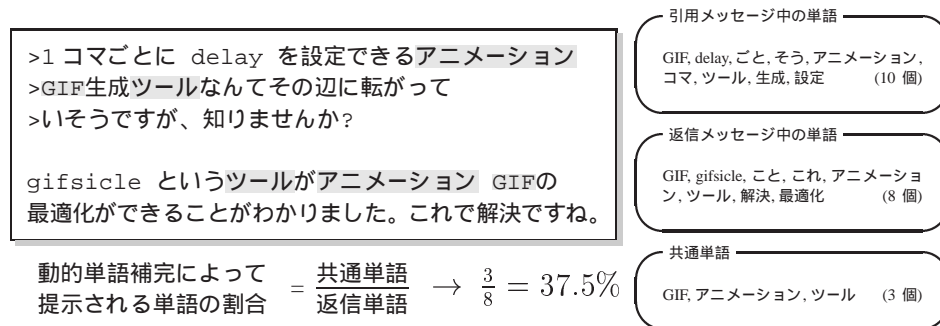


図 6. 評価の例

実際の評価は 6 つのメーリングリストから返信メールを抽出して行った. メーリングリストの種類はソフトウェア開発のコミュニケーション (開発者 1, 2), ソフトウェア利用者間のコミュニケーション (ユーザ 1, 2), 友人同士のジャンルを問わないコミュニケーション (友人 1, 2) を 2 つずつ選択した. その結果, どのメーリングリストもお

⁶ たとえば, 「コンテキストを考慮した動的単語補完による入力効率の向上」という文では, 「を」「した」「による」「の」といった助詞などの用言はコンテキストの形成には寄与しない.

よそ平均 10% 前後の単語をコンテキストから抽出して提示できることが分かった。図 7 に各メーリングリストの提示率の分布を箱髷図で、平均提示率を表で示す。

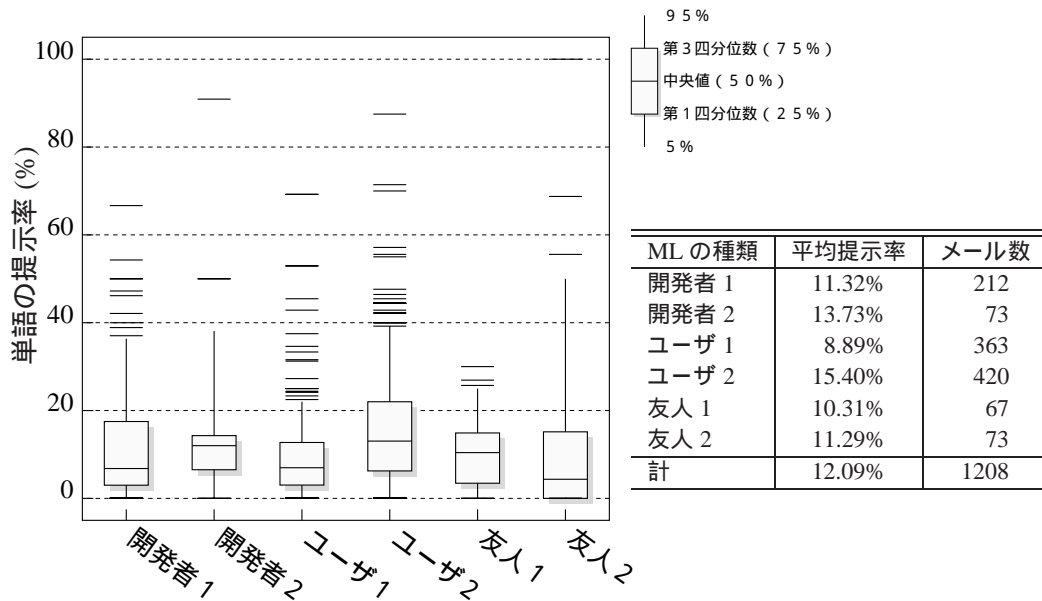


図 7. 単語の提示率

図 8 に示すように、高い提示率を示したメールは特定の話題について議論を行う形式が多く見られた。このような場面では動的単語補完が効果的に活用できる。対して、低い提示率であったメールは短い文章のメールであったり、話題が変化しているメールが多く含まれていた。この原因は、そもそも動的単語補完が捉えるべきコンテキストが引用メッセージの中に存在していないからである。表 1 に示す通り、予測候補の提示率が低いメールには引用メッセージ中に含まれる単語数が少ない。

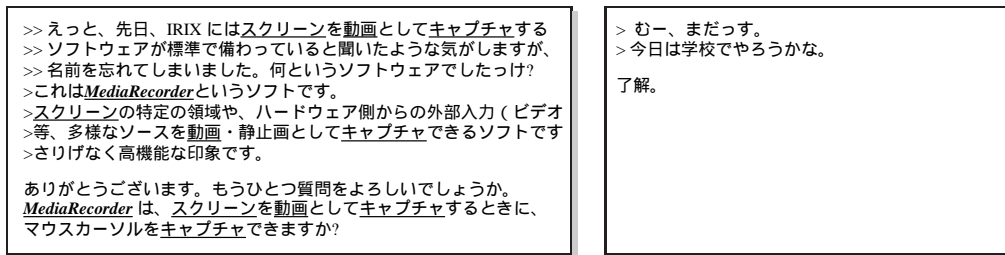


図 8. 効果的な例(左図)と効果的ではない例(右図)

6 議論

文字の入力と検索は極めて近い関係にあると言える。かな漢字変換は、日本語の読みから漢字への変換を、読みと漢字が対応した辞書を検索することによって実現されている。かな漢字変換システムなどの文字入力システムが、与えられた文字列に対して

MLの種類	全体		提示率 5% 未満		提示率 5% 以上	
	メール数	平均単語数	メール数	平均単語数	メール数	平均単語数
開発者 1	212	23.3	80	14.4	132	28.7
開発者 2	73	20.6	11	16.2	62	21.4
ユーザ 1	363	17.4	135	9.8	228	21.8
ユーザ 2	420	21.4	90	11.8	330	24.0
友人 1	67	23.9	19	13.2	48	28.2
友人 2	73	13.0	33	9.2	40	16.1

表 1. 提示率と引用メッセージに含まれる平均単語数の関係

静的な辞書を検索するだけでなく、周辺の関連文書も動的に検索すれば、よりコンテキストに適した文字入力を実現できる。

我々が今回提案した予測入力方式における動的単語補完の活用では編集中心の文書のコンテキストのみを利用したが、将来的には利用者の保有しているすべての文書をコンテキストとして利用可能な入力方式の作成を目指している。

編集中心の文書のコンテキストを利用した他のシステムとしては、Remembrance Agent[2]が挙げられる。Remembrance Agentでは、編集中心の文書のコンテキストを元にエージェントが率先して情報検索を行い、利用者に有益と思われる情報を提示する。

入力の効率化を計る我々の予測入力システムと、記憶力を補助する情報検索システムである Remembrance Agent では目的が異なるが、入力システムの検索機能を強化すれば、過去に作成または参照したすべての文書から必要な情報を取り出すという情報検索の用途にも適用できる。我々は、入力と検索をなめらか統合し、蓄積された文書資産を有効に再利用できる予測入力システムの実現を目指している。

7 まとめ

我々はコミュニケーションのテキスト入力におけるコンテキストの重要性に着目した。そして、予測入力方式の候補に、日本語動的単語補完の手法である Nanashiki によって得られた単語を加えて提示することにより、コンテキストを活用した予測入力方式を実現した。今後、メールやチャットなどの電子的なコミュニケーションがより日常的になるにつれて、我々が提案したコンテキストを捉えた予測入力システムは、ますます重要になると考えている。

参考文献

- [1] T. Masui. An efficient text input method for pen-based computers. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '98)*, pp. 328–335. Addison-Wesley, April 1998.
- [2] B. J. Rhodes. Remembrance agent. In *The Proceedings of The First International Conference on The Practical Application Of Intelligent Agents and Multi Agent Technology*, pp. 487–495, 1996.
- [3] 井田昌之, 亀井信義. Emacs 解剖学 入力の補完. *bit*, Vol. 29, No. 2, pp. 85–95, 1997.
- [4] 高林哲, 小松弘幸, 増井俊之. Migemo: 日本語のインクリメンタル検索. 情報処理学会研究会報告 2001-HI-94, pp. 41–46, July 2001.
- [5] 松本裕治, 北内啓, 山下達雄, 平野善隆, 松田寛, 高岡一馬, 浅原正幸. 日本語形態素解析システム『茶釜』 version 2.2.0, 2000. <http://chasen.aist-nara.ac.jp/>.