インターフェイスの街角 (66) — 状況依存の予測入力

増井 俊之

予測型テキスト入力は、PDA や携帯電話ではおなじみの手法になりつつあります。ここ数年、私は UNIX 上の Emacs や Windows アプリケーションなどでも予測型入力手法である POBox を使用しています。POBox のような予測型入力システムでも一般的なかな漢字変換システムでも、どの単語を確定したかによって学習がおこなわれるのが普通なので、1 回使った単語や文章は 2 回目以降は入力しやすくなるという特徴があります。このため、使っていくにつれて自動的にカスタマイズがおこなわれることになります。しかし、初めて使う単語の場合は予測や変換がうまくいかない場合があります。そもそも、どんな状況でも同じ手法で予測や変換がおこなわれるのは適切な手法ではないかもしれません。

文章を書くときの状況に応じて予測の手法や辞書を切り替えることができれば、入力の効率が向上する可能性があります。ユーザーの違いだけでなく、利用するアプリケーションや入力をおこなう時間や場所、場合によっては気分によって辞書や予測手法を切り替えられると便利でしょう。

このように、予測入力における辞書の切替え手法はいろいろと考えられますが、今回は、入力しようとしているテキストに関連する既存の文書を利用して予測辞書を切換える方法について考えてみます。

動的単語補完による状況依存入力

予測型入力システムを用いて計算機についてのメールを書いている場合、〝kon〟と入力したときには〝コンピュータ〟や〝コンパイラ〟などの単語が予測されると好都合です。しかし、コンサートへの誘いのメールに対して返事を書いているのなら、〝コンサート〟と予測されるほうが適切かもしれません。

コンサートへの誘いのメールには、〝コンサート〟という文字列が含まれている可能性が高いと思われます。したがって、もとのメールに含まれる単語をもとに入力予測をおこなうことにすれば、〝kon〟から〝コンサート〟を予測させることができるはずです。この場合は、返事の対象になるメールのテキストを予測辞書として使えばよいことになります。

動的単語補完

既存のテキストを入力に活用する手法は〝動的単語補完 (dabbrev:Dynamic Abbreviation)〟と呼ばれ、Emacs 上で使用することができます。たとえば、Emacs でこの記事を書いているとき、〝Abb〟と入力してから M-/キーを押すと、動的単語補完をおこなう dabbrev-expand 関数が呼び出され、〝Abb〟が〝Abbreviation〟に展開されます。

dabbrev-expand は、編集中のテキストのなかから入力中の文字列と同じプレフィックス (接頭辞) をもつ単語を検索し、みつかった場合は入力中の文字列をその単語に展開します。上の例では、Abbreviation という単語がテキストに含まれているため、Abb が Abbreviation に展開されますが、テキストに〝Abbey〟のような単語も含まれている場合は Meta-/キーを押すたびに Abbreviation、Abbey、Abb のように候補単語が切り替わっていきます。

POBox などの予測入力システムにおいても、予測辞書にもとづく候補だけでなく、dabbrev による展開結果も候補として表示すれば、文中にある既出の単語も入力単

語として選べるようになります。このようにすれば、任意のテキスト全体を予測辞書として使えるようになるわけです。

日本語の動的単語補完

動的単語補完は単独で使っても、予測入力に応用しても便利なものですが、日本語の入力での利用は簡単ではありません。

まず、日本語テキストは単語の区切りが自明でなく、文面を見ただけではどこからどこまでを単語とみなせばよいのかが判定できません。また、単語の区切りが明白であったとしても、動的単語補完を利用して `日本語´のような単語を選択するために `日´という文字を入力しなければならないとすると、入力効率の向上にはほとんど役に立たないでしょう。このように、英語版の dabbrev のアルゴリズムを単純に日本語テキストに適用することは無意味といっても過言ではありません。

東京工業大学の小松浩幸氏らは、日本語の動的単語補完を実現した「七色」というシステム[1]を考案しました[1]。七色は、上記の問題に次のような方法で対処しようというものです。

1. 形態素解析システムで単語の区切りを検出する
   これまでにも何回か紹介しましたが、「茶筌」[2]や「MeCab」[3]などの形態素解析プログラムを使うと、かなり正確に日本語テキストを品詞に分解することができます。その結果を解析して適当につなげば、動的単語補完で使うべき単語を構成することができます。
2. ローマ字で日本語単語を検索できる「Migemo」[2]で単語検索をおこなう
   前述のように、`日´という文字の `日本語´という単語への展開にはあまり意味はありませんが、`ni´を `日本語´に展開できるのなら重宝しそうです。2002年 2 月号の「横着プログラミング」で紹介されていた Migemo システム[4]を使えば、ローマ字で日本語単語を検索できます。つまり、形態素解析によって分解された単語を Migemo で検索すれば、展開の候補となる単語がみつかることになります。

1 http://www.taiyaki.org/pobox/nanashiki.html
2 http://chasen.aist-nara.ac.jp/
3 http://cl.aist-nara.ac.jp/~taku-ku/software/mecab/
4 http://migemo.namazu.org/

図 1　動的単語補完をおこなう前の状態

```
* ドーナツとトンカツ
* Read The Document
* 単語の動的略語展開
do

MULE/7bit--**-XEmacs: *scratch*
```

図 2　七色による日本語動的単語補完を適用

```
* ドーナツとトンカツ
* Read The Document
* 単語の動的略語展開
動的略語展開

MULE/7bit--**-XEmacs: *scratch*
[動的略語展開] Document　ドーナ
```

```
* ドーナツとトンカツ
* Read The Document
* 単語の動的略語展開
Document

MULE/7bit--**-XEmacs: *scratch*
[Document] ドーナツ　動的略語展
```

七色の使用例

図 1 のような日本語テキストで `do´と入力してから七色を適用すると、まずカーソルの前のテキストが `ドーナツ/と/トンカツ/Read/The/Document/単語/の/動的/略語/展開´のように分解されます。その後、読みが `do´にマッチする単語が Migemo によって検索され、`動的略語展開´や `Document´などに展開されます (図 2)。

このような日本語動的単語補完機能を POBox などの予測入力機能と組み合わせれば、`do´と入力した時点で `動的´や `Document´ `ドーナツ´などを予測入力の候補に含めることができます。最初のほうに挙げた例でいえば、コンサートについてのメールに返事を書いているときは `kon´という入力に対して `コンサート´という候補が示され、計算機に関連するメールを書いている場合は `コンピュータ´が候補になるわけです。

類似文書による状況依存入力

動的単語補完を予測入力に利用する場合、テキストに含まれる単語を候補にすることはできます。しかし、いくら関係の深い単語であっても、そのテキストに含まれていなければ予測には使えません。たとえば、

今度の日本フィルのコンサートに一緒に行きませんか?

というメールに返事を書く場合、〝日本フィル〟や〝コンサート〟という単語は予測入力の候補になりますが、コンサート会場などは候補から外れてしまいます。しかし、

昨日、サントリーホールの日本フィルのコンサートに行ってきました。

というメールが別にあるのなら、このテキストを予測に使い、〝san〟という入力文字列から〝サントリーホール〟を予測させることも可能になります。このように、ある文章を入力するとき、それに類似した文章を予測辞書として活用することによって予測効率の向上が図れる可能性があります。

2002 年 12 月号と 2003 年 3 月号で、汎用連想計算エンジン GETA (Generic Engine for Transposable Association)[5]を用いた全文検索システムと類似ファイル検索システムを紹介しました。文章を入力するときはかならずリアルタイムに類似ファイルを検索し、みつかったファイルに書かれている単語を予測に使うことにすれば、動的単語補完を利用する場合よりも効果的な予測が可能になるのではないでしょうか。

GETA による検索処理では、ある単語があるファイルでいくつ使用されているかを表現するインデックスが作成され、それをもとに単語の重要度を計算してファイル検索がおこなわれます。各ファイルにおける単語の使用状況がすでにデータベースとして格納されているわけですから、これを辞書とみなしてアクセスすることにより、簡単に予測辞書として使うことができます。

類似変換の実装

七色を用いた動的単語補完では、予測するたびに、辞書として利用する参照テキストを形態素解析して単語列に分割する必要があるので、処理に時間がかかってしまいます。一方、GETA のインデックスはすでに単語に分割された状態で構築されているため、形態素解析を実行する必要はありません。

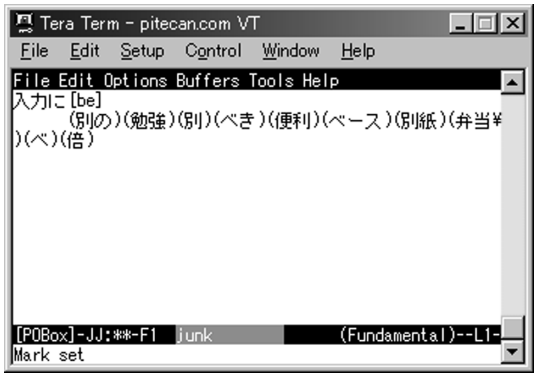
さきほどの例では、

今度の日本フィルのコンサートは

と入力した時点で、この文字列から類似文書検索を実行すると、

5 http://geta.ex.nii.ac.jp/

図 3　初期状態で〝be〟と入力したときの候補表示



昨日、サントリーホールの日本フィルのコンサートに行ってきました。

という文書がみつかります。ここで GETA のインデックスを参照し、この文書に含まれている〝昨日〟〝サントリーホール〟〝日本フィル〟〝コンサート〟などの単語が得られます。そして、これを予測辞書として使い、〝san〟という入力から〝サントリーホール〟を予測する仕組みです。

POBox サーバー上の実装例

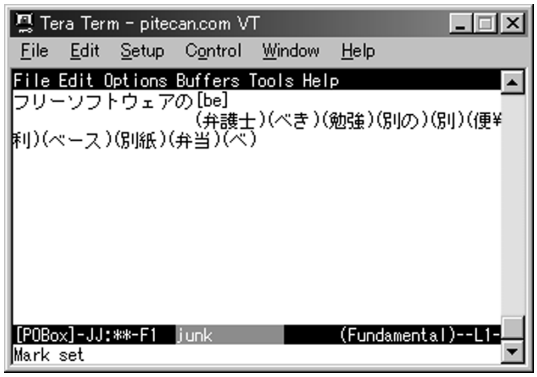
ここまでに述べた方針にもとづいて、POBox サーバーを実装してみました。本来なら、入力中のテキストをもとに類似ファイルを検索し、辞書として利用すべきでしょう。しかし、今回は簡便化するため、POBox で最近確定されたいくつかの単語をもとに全文検索をおこない、その文書に含まれる単語を予測辞書として利用することにしました。上記の例でいえば、〝今度〟〝日本フィル〟〝コンサート〟と順番に確定したあとで類似検索をおこなうと、これらの単語と〝サントリーホール〟を含む文書がみつかります。

これを実装した POBox サーバーと、私の手許にあったメールやファイルを用いて、どのように変換がおこなわれるかを実験してみました。

図 3 は、私が使っている Emacs 版 POBox の初期状態で〝be〟と入力した直後の様子です。〝別の〟や〝勉強〟など、一般的な単語が候補として表示されていることが分かります。

一方、図 4 は、〝フリー〟と〝ソフトウェア〟という単語を確定してから〝be〟と入力したときの様子です。たまたま、そのすこし前に開かれた著作権関連の会合における

図 4　〝フリー〟〝ソフトウェア〟を確定後に〝be〟と入力したときの候補表示



Richard Stallman 氏の講演のレポートをメールで受け取っていたため、〝フリー〟や〝ソフトウェア〟というキーワードからこのレポートが検索され、そのなかに含まれる〝弁護士〟のような単語が優先的に候補として表示されています。

おわりに

状況に応じて効率よくテキスト入力をおこなう方法は便利だということは分かっていても、これまでは簡単に実装するのが難しかったため、いわば〝絵に描いた餅〟で、あまりひろく使われていませんでした。しかし、全文検索システムなどとうまく組み合わせれば、案外早く実用化への道が開けるのではないかと思います。

一般に入力と検索のシステムは別物とみなされていますが、今回の例のように統合的に考えれば、入力と検索の両方に有用なシステムを作れそうです。

入力用の辞書の学習結果をみれば、入力したテキストの内容まで想像がついてしまうことがあるため、辞書の扱いには注意が必要です。しかし、辞書を自由自在に切り替えて使えるのなら、このような危険性を減らすこともできるかもしれません。柔軟で動的な辞書の応用をもっと考えてみたいと考えています。

（ますい・としゆき　産業技術総合研究所）

[参考文献]

[1] 小松弘幸、高林 哲、増井俊之「日本語動的単語補完方式 Nanashiki を活用した予測入力」、『インタラクティブシステムとソフトウェア IX：日本ソフトウェア科学会 WISS2001』、pp. 67-74、近代科学社、2001 年 12 月

[2] 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」、情報処理学会論文誌、Vol.43、No.12、pp.3,698–3,705、2002 年 12 月