意味処理に基づく入力予測辞書の構築および入力支援インタフェース

大倉清司、長瀬友樹、潮田明
(株) 富士通研究所
{okura.seiji, nagase.tomoki, ushioda}@jp.fujitsu.com

1. はじめに

日常業務や日常生活において、テキスト入力の効率化が求められている。従来技術としては、ＰＣ上のかな漢字変換や、携帯電話上の予測入力機能、ブラウザ上のURLのオートコンプリート機能などが挙げられるが、文例の単位が短すぎたり長すぎたりするため、入力はあまり効率化できない。今回、過去の入力から、表記は違っても同じ意味を表す文字列を表現単位で抽出し、頻度順に並び替えて入力予測辞書を抽出することに成功した。適当な長さの予測候補の抽出およびその意味計算をすることにより、効果的な予測候補が抽出できる。また文例辞書を使い、既存のかな漢字変換と連携して入力支援をするＧＵＩをＰＣ上で開発した。本稿では、入力予測辞書を抽出する技術およびその実装について説明する。

2. 従来の入力予測技術とその課題

従来から、ユーザが入力したいと予想される文字列候補を表示し、その中から選択させることにより入力を効率化する技術はあった。ユーザの入力を予測するために、過去に入力された文字列から繰り返し使われている文字列（以下、文例と呼ぶことにする）をどのように辞書化するかが技術のポイントになる。従来技術には以下のようなものがあった：

1.　ＰＣ上のかな漢字変換

入力予測技術の筆頭としては、ＰＣ上のかな漢字変換[1,2,3]が挙げられる。ユーザがかなを未確定状態で入力して変換キーを押すと、漢字変換候補が表示される。はじめは誤った候補が表示されるかもしれないが、ユーザが入力するにつれ、入力内容を学習し、次第に入力したい候補が上位に表示されるようになる。通常は辞書機能もあり、よく使う文字列をあらかじめ登録しておくこともできる。

2.　携帯電話上のかな漢字変換

ＰＣに比べて文字入力の負荷が高い携帯電話においては、古くから予測入力機能があった[4]。ＰＣ上のかな漢字変換技術をベースに携帯端末に適用したものである。変換キーを押さなくても、未確定状態で、確定文字列を予測する機能や、漢字変換の確定時に、過去の入力履歴から次の確定候補を予測する機能などがある。これにより、携帯電話での入力効率が大幅に改善された。

3.　オートコンプリート技術

ブラウザ上でURLを直接入力したい場合、以前に同じURLにアクセスしている場合にはURLの一部を入力するだけで、それにマッチした過去のURL候補を表示する機能がある。またメールソフトには、メールアドレスの先頭数文字を入力するだけで、それに前方一致するメールアドレスを自動補完したり、ニックネームを入れることによりメールアドレスを表示する機能がある。これらの技術により、入力の手間が大幅に軽減された。表計算ソフトで、過去に入力した文字列をオートコンプリートする技術もある。また、翻訳支援システムにおいて訳文の入力負荷を軽減する技術[5]などもあった。

しかし、これらの技術は以下の点で問題があった：

1. 表示される候補が短すぎる場合には、何度も確定キーを押して入力を続ける必要があった。
2. 表示される候補が長すぎる場合には、一度候補を選択して確定してから、確定文字列を削除または編集する必要があった。
3. 従来技術では確定のタイミングで候補が記憶されるため、通常は1文節以上の長さで候補が自動学習されることはなかった。

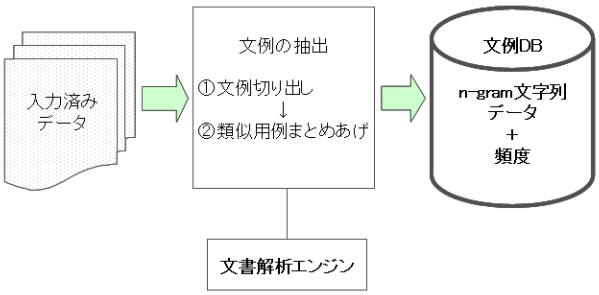

図 1　文例抽出の概要

文節以上の単位で候補を出したいときは、ユーザ自身で文例の登録作業をする必要があった。

4. 同じ意味を表す表現でも、表記が違うと別の候補として記憶されてしまっていた。例えば、コンピュータの稼動状況を記録するとき、「動作:正常」（コロンは半角記号）「動作：正常」（コロンは全角記号）、「動作は正常」、「動作正常です」のように様々な表現がある。入力効率の向上のためには、文例を頻度順にソートして表示すると効果的だが、このように同じ意味でも表記が違う場合は、頻度が正確にカウントできないという問題も起きる。

これらの問題に対処するには、適当な長さの文例候補を自動抽出し、同じ意味を表す文例候補をまとめて頻度をカウントする必要がある。今回、この要件に対処するために、以下 2 つの解決策を考案した。

(1)適当な長さの文例を自動で抽出するために、形態素解析技術を使い、複数文節からなる文例候補を抽出する
(2)同じ意味を表す文例候補を 1 つにまとめるために、文例候補に対して意味処理を行う

3. 意味処理に基づく入力予測辞書の構築

本技術は、過去の入力に基づき文例を自動的に辞書化する。かな漢字変換のときに文例を提示することにより、入力を効率化する。例えば未確定状態で「どうさ」と入力すると、「動作:正常」「動作:異常」などの文例候補を一覧表示する。ユーザはその中から入力したい文例を選択することで簡単に入力を行うことができる。

表 1　過去の入力例とその解析結果例

過去の入力	***解析結果例*（左側は形態素、右側は意味記号）**
動作:正常	**動作**　OPERATE : **正常**　NORMAL
動作は正常です	**動作は**　OPERATE **正常です**　NORMAL
動作正常	**動作**　OPERATE **正常**　NORMAL
動作:異常	**動作**　OPERATE : **異常**　ABNORMAL

文例抽出の大まかなフローを図 1 に示す。過去の入力に対して、文書解析エンジンを使い、①文例の切り出しと②類似用例のまとめあげを行う。

3.1. 文例の切り出し

提示される文例候補は、短すぎると検索回数が多くなり、長すぎても文例確定後の編集が必要になるため、入力効率が下がってしまう。過去の入力を効率化するために、文節の単位を文例の最小単位とし、文節を適当な長さに連結させたものを文例候補とすることにした。このために形態素解析処理を行い、文節合成を行った。

通常、文節とはそれだけで意味がある単位をさすが、本稿における文節は、記号も 1 文節と見なすことにする。例えば、「動作：正常」という入力は「動作」「：」「正常」の 3 文節からなる。

文例の切り出しの処理について説明する。まず、過去の入力に対して文を抽出し、それぞれの文に対して形態素解析処理を行う。表 1 は過去の入力とその形態素解析結果の例である。例えば、「動作は正常です」を入力とした場合、「動作」「は」「正常」「です」という形態素に分割される。このとき、各形態素に対して、自立語に対しては意味記号を抽出する。例えば「動作」の意味記号は”OPERATE”、「は」は非自立語なので意味記号はなく、「正常」の意味記号は”NORMAL”、「です」は非自立語なので意味記号はない。

表 2 2-gram 以上の文例候補を作成した例

入力	*2-gram の例*	*3-gram の例*
動作:正常	動作／: :／正常	動作／:／正常
動作は正常です	動作は／正常です	－
動作正常	動作／正常	－
動作:異常	動作／: :／異常	動作／:／異常

次に、形態素解析結果から文節を合成する。このとき、意味記号の情報も合わせて合成し、記憶しておく。これは後の処理で使う。例えば、「動作」＝自立語、「は」＝非自立語（助詞）であるので、文節合成を行い「動作は」を得る。この文節を表す記号は、“OPERATE”となる。同様に、「正常」＝自立語、「です」＝非自立語を合成して、文節「正常です」を得る。この文節を表す記号は“NORMAL”となる。このとき、1 文節に複数の意味記号があるときは、意味記号をアルファベット順にソートして連結する。

次に、文節を複数つないだもの(n-gram,n=2～9)を文例候補とする。なお、文末がきたらそれ以上つなげない。表 2 は、表 1 によって得た文節を 2 文節、3 文節とつないだものである。「／」は文節間の区切りを表すための便宜的な記号である。この各文例候補に対して頻度を計算する。

文節数については、当初は 4～5 文節からなる文例が有効であると考えていた。しかし、実際のデータで試行したところ、2 文節のものも有用で、9 文節のものも効果的なことがわかった。文節数を変えたシステムを複数作成し、想定される使用者にヒアリングを行うという検証を繰り返し、最終的に文節数は 2～9 が適切なことがわかった。

表 3 は表 2 における各文例候補の n-gram 頻度を表したものである。各文例候補に対する頻度をもとに、以下の処理を行う：

- n 文節からなる文例候補(A)の頻度と、(A)を前方一致部分文字列として含む(n+1)文節からなる文例候補(B1)，(B2)，…の頻度を合計したものが同じ場合、文例(A)を削除する

表 3 文例候補の n-gram 頻度

2-gram文字列	*頻度*
動作／:	2
:／正常	1
動作は／正常です	1
動作／正常	1
:／異常	1

3-gram文字列	*頻度*
動作／:／正常	1
動作／:／異常	1

4-gram文字列	*頻度*
:	:

例えば、2-gram 文節「動作／：」（頻度＝2）を 3-gram の文例候補から前方一致検索すると、「動作／：／正常」（頻度＝1）と「動作／：／異常」（頻度＝1）を得る。これら 2 つの 3-gram 文例候補の頻度の合計は 2 である。これは、「動作／：」と入力された場合には必ず「動作／：／正常」または「動作／：／異常」と入力されることを表している。つまり、「動作／：」は文例候補としては短すぎるのである。そこで「動作／：」を削除する。この削除処理を n=2～8 まで順次行うことにより、短すぎる文例を削除することができる。

また、実際のデータによる試行の結果、文の途中からの文例は削除することにした。図 4 中でアンダーラインで示される「動作／：」「：／正常」「：／異常」は、以後の処理対象にはならない。

3.2. 類似用例まとめあげ

過去の文例から同じ意味の表現をグループ化し、それぞれ使用頻度を集計することにより、使用頻度の高い表現を入力候補の上位にランク付けする技術を開発した。これにより、表記は違うが同じ意味を表す類似した文例が多く表示されるのを防ぎ、文例選択の時間が短縮される。また表記を統一する効果もある。

例えば表 3 で同じ意味を表す文例候補は

動作は／正常です
動作／正常
動作／：／正常

の 3 つである。

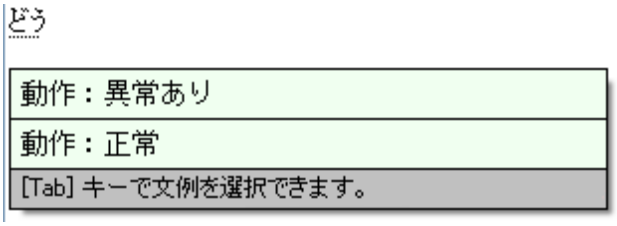

図 2　文例候補表示の例

各文例候補が同じ意味かどうかは、各文節候補に対して文節を表す意味記号をアルファベット順にソートし、その結果を連結した文字列（意味表記と呼ぶ）を比較することにより判断できる。例えば、「動作は」の意味記号は”OPERATE”、「正常です」の意味記号は”NORMAL”となり、「動作は／正常です」の意味表記は”NORMAL_OPERATE”となる。同様にして「動作／正常」は”NORMAL_OPERATE”、「動作／：／正常」は”NORMAL_OPERATE”となり、全て同じ意味表記となっている。なお、「動作／：／異常」は”ABNORMAL_OPERATE”となる。

同じ意味を表す文例候補が複数あった場合には、代表的な文例 1 つを抽出する必要がある。同じ意味を表す文例候補のうち、頻度が最大のものを抽出することにした。頻度がどれも同じ場合は、最長文字列を代表的な文例とした。また、代表的な文例の頻度は、同じ意味を表す文例候補の頻度の総計とした。この結果、表記が違うと頻度がばらけてしまう問題が解決され、代表的な文例に、同じ意味のフレーズ全体の頻度を付与することができた。

この処理を行うことにより、以下の文例が抽出される：

動作は正常です（頻度＝3）
動作：**異常**（頻度＝1）

最後に、検索のために文例と一緒に文例の読みも辞書に保存する。

4.　入力予測システムの試作

本稿で提案した手法で抽出した文例を活用して、入力予測をするＧＵＩを開発した。従来のかな漢字変換の操作性を損なわず、シームレスに文例検索機能と連携させた。

図 2 は今回の入力システムの画面例である。「どう」と未確定文字列を入力したところで、文例が自動的に検索され、頻度順に表示される。上の例だと、「動作が正常です」の頻度が 3、「動作：異常」の頻度が 1 であるので、頻度がより高い「動作が正常です」が最初に表示される。

5.　今後の課題

この技術を使い、メール文・電子カルテの入力などで実証実験を行い、評価したい。

今後の課題は 2 つある。1 つ目は、同じ意味を表す類似文例のまとめ処理に関して、現状では対処できない現象について研究することである。例えば「動作：正常」「動作：異常なし」は違う意味の文例と計算されてしまう。これは、「正常」の意味記号が”NORMAL”、「異常」の意味記号が”ABNORMAL”であるため、”ABNORMAL”を否定しても、”NORMAL”とならないためである。この問題に対処するには、シソーラスなどが必要となる。

2 つ目は、文例同士の非対称性を解消することである。例えば「動作：異常」の反対の意味の文例として「動作：正常」を抽出したいが、単純な文例頻度で計算すると「動作は正常です」が抽出されてしまう。「動作は異常です」という過去の入力例がない場合は、「動作は正常です」は候補とするのは不自然である。文例同士の非対称性に対処するためにも、シソーラスの活用が必須になると考える。

参考文献

[1]小林龍生．漢字・日本語処理技術の発展：仮名漢字変換技術．情報処理学会学会誌 Vol. 43 No. 10, 2002.
[2]市村由美, 齋藤佳美, 木村和広, 平川秀樹. 予測に基づく入力支援機能を備えたかな漢字変換システムの開発. 自然言語処理 Vol.129, No.10, 1999.
[3]田中久美子, 早川大地, 武市正人, 玉井哲雄. ユーザ文書を用いた個別かな漢字変換支援. 自然言語処理 Vol.152, No.22, 2002.
[4]T. Masui. POBox: An efficient text input method for handheld and ubiquitous computers. In *Proceedings of the International Symposium on Handheld and Ubiquitous Computing (HUC'99)*, 1999.
[5]大倉清司, 富士秀, 長瀬友樹．オートコンプリートによる翻訳支援．言語処理学会第 13 回年次大会予稿集, 2007.