日本語形態素解析システム **JUMAN**
使用説明書　**version 2.0**

松本裕治　黒橋禎夫　宇津呂武仁　妙木 裕　長尾 真

平成 6 年 7 月

Japanese Morphological Analysis System JUMAN Manual

Version 0.6	17 February 1992
Version 0.8	14 April 1992
Version 1.0	25 February 1993
Version 2.0	11 July 1994

目 次

1 はじめに **1**

2 日本語形態素文法 **2**

2.1 形態品詞および品詞細分類 2

2.2 活用型と活用形 2

2.3 形態素構造 2

2.4 連接規則と連接関係 3

3 辞書の定義とデータ構造 **3**

3.1 辞書の概要 3

3.2 文法辞書の定義法 4

3.3 形態素辞書の記述法 5

3.4 システム辞書の作成 6

3.4.1 連接規則辞書の変換 7

3.4.2 中間辞書への変換 7

3.4.3 システム形態素辞書への変換 7

3.5 複数辞書の使用 8

3.6 システム標準文法 8

3.7 オプション定義 (cf. .jumanrc) 9

4 形態素解析 **11**

4.1 形態素解析アルゴリズム 11

4.2 未定義語の取り扱い 12

4.3 C 版 12

4.4 Prolog 版 13

参考文献 **14**

付録 **16**

A ユーザ辞書の記法の定義と例 **16**

A.1 形態品詞分類辞書 (JUMAN.grammar) の記述 16

A.1.1 形態品詞分類辞書の個々の項目の記述 16

A.1.2 形態品詞分類辞書の定義例 16

A.2 活用関係辞書 (JUMAN.kankei) の記述 17

A.2.1 活用関係辞書の個々の項目の記述 17

A.2.2 活用関係辞書の定義例 17

A.3 活用辞書 (JUMAN.katuyou) の記述 18

A.3.1 活用辞書の個々の項目の記述 18

A.3.2 活用辞書の定義例 18

A.4 連接規則辞書 (**JUMAN.connect**) の記述 19

A.4.1 連接規則の記述 19

A.4.2 連接規則の定義例 19

B 形態素辞書の記述 **20**
B.1 形態素の記述 20
B.2 形態素の記述例 20

C オプションファイル (`.jumanrc`) の記述例 **21**

D インストレーション手順 **22**
D.1 システムの生成 22
D.2 システム辞書の生成 22

E JUMAN 0.8 から JUMAN 1.0 への拡張点 **23**

F JUMAN 1.0 から JUMAN 2.0 への拡張点 **23**

1　はじめに

計算機による日本語の解析において，欧米の言語の解析と比べてまず問題になるのに次の 2 点があります．一つは形態素解析の問題です．ワードプロセッサの普及などによって日本語の入力には大きな問題がなくなりましたが，計算機による日本語解析では，まず入力文内の個々の形態素を認識する必要があります．これには実用に耐えられるだけの大きな辞書も必要であり，これを如何に整備するかという問題も同時に存在します．もう一つの問題として，日本語には広く認められ同意を得られた文法，ないし，文法用語がないという現実です．学校文法の単語分類および文法用語は一般には広く知られていますが，研究者の間ではあまり評判がよくありませんし，計算機向きではありません．

日本語の解析に真っ先に必要な形態素解析システムは，多くの研究グループによって既に開発され技術的な問題が洗い出されているにも係わらず，共通のツールとして世の中に流布しているものはありません．計算機可読な日本語辞書についても同様です．

本システムは，計算機による日本語の解析の研究を目指す多くの研究者に共通に使える形態素解析ツールを提供するために開発されました．その際，上の二つ目の問題を考慮し，使用者によって文法の定義，単語間の接続関係の定義などを容易に変更できるように配慮しました．

大学で小人数で開発したシステムであり，色々な点で不完全な部分があると思います．可能な限り順次改良を重ねる予定です．皆様の寛容な利用をお願いいたします．

本システムは京都大学長尾研究室の多くの学生たちの協力を得て作成したものです．また，辞書に関しては，長尾研究室で行なわれた Mu プロジェクトで開発された辞書， Wnn かな漢字変換システムの辞書，および， ICOT から公開された日本語辞書を利用いたしました．また，米国 MCC の自然言語グループの方々には Version 1.0 でハッシングによる辞書引きの高速化を実現していただきました． SICStus Prolog とのソケットによる接合部を東京工大の徳永さんに作成していただきました．また，富士通の中村俊久さんには ndbm による辞書のハッシュデータベース化を実現していただきました．御協力いただいた各位にお礼申し上げます．

平成 6 年 7 月

本システムに関するお問い合わせは以下にお願いします．
〒 630-01
奈良県生駒市高山町 8916-5
奈良先端科学技術大学院大学
情報科学研究科
松本裕治
Tel: (07437)2-5240
Fax: (07437)2-5249
Email: matsu@is.aist-nara.ac.jp

2 日本語形態素文法

本システムで仮定している日本語の形態素文法およびここで用いる用語について説明する．

2.1 形態品詞および品詞細分類

本システムでは形態素の集合の二階層の分類が仮定されている．これらの分類を形態品詞および (形態) 品詞細分類と呼ぶ．

例えば，名詞を普通名詞，サ変名詞，固有名詞などに細分類することができる．また，ある形態品詞を細分類したくない場合は細分類を指定しなくてもよい．例えば，品詞細分類を持たない形態品詞として動詞を定義してもよい．

この二階層に分類されたそれぞれの集合を品詞と呼ぶ．上の例では，名詞の細分類としての普通名詞，サ変名詞，固有名詞，および，動詞はそれぞれ品詞である．

2.2 活用型と活用形

学校文法における '動詞' '形容詞' '形容動詞' などのように，後接する形態品詞に応じてその形態が変化する形態素が存在する．このような形態変化を活用 (conjugation) という．このような形態素の表層のうち変化しない部分を (活用) 語幹，変化する部分を (活用) 語尾と呼ぶ．大部分の活用は規則的である．その規則性に従って活用のタイプを分類し，これを活用型と呼ぶ．そして，連接に応じて実際にとり得る表層的な個々の形態を 活用形と呼ぶ．

2.3 形態素構造

形態素 m の形態品詞，品詞細分類，活用型，活用形，表層形がそれぞれ $H1, H2, K1, K2, M$ であるとき，これを 5 項組 $(H1\ H2\ K1\ K2\ M)$ によって表わし，「形態素 m の 形態素構造」と呼ぶ．

形態素構造の項 $H1, H2, K1, K2$ には，具体的な名称として，それぞれ形態品詞名・品詞細分類名・活用型名・活用形名を与える．また，項 M には，漢字または仮名で表記される表層形を与える．各形態素構造の記述には特別なシンボル '$*$' を使用することができる．'$*$' は，その項を考慮しない (未指定，'don't care') ことを表わす．形態素構造を表わすリストのある要素以降がすべて未指定でよい場合は，形態品詞の部分以外はそれらを省略してもよい．たとえば，形態素構造

$$\alpha_1 = (*)$$

は任意の形態素を表わす集合とみなすことができる．同様に，形態素構造

$$\alpha_2 = (\text{名詞})$$

は '名詞' という形態品詞に分類されたすべての形態素の集合を表わす．また，形態素構造

$$\alpha_3 = (*\ *\ *\ \text{未然形})$$

は '未然形' という活用形を取るすべての形態素の集合を表わす[1]．また，

$$\alpha_4 = (\text{助詞格助詞}\ *\ *\ \text{が})$$

のように具体的な形態素を指定することも許される[2]．

この記法は次に述べる連接規則を記述する際に使われる．

[1] この例では，$\alpha_1 \supseteq \alpha_2, \alpha_1 \supseteq \alpha_3$ という関係が成り立つ．

[2] この例ように，活用しない形態素に対しても 5 項組によって形態素を表現する必要があり，表層形を指定する場合には活用型名，活用形名の部分を $*$ としなければならない．

2.4 連接規則と連接関係

形態素構造の集合の組 A_1 と A_2 に含まれるそれぞれの要素が文中に連続して現れ得るとき，それらは連接可能であるといい，それを記述した規則を連接規則と呼ぶ．また，連接規則によって定義される 2 項関係を連接関係と呼ぶ．

連接規則は二種類の形態素の連接可能性が記述可能である．また，連接の強弱に関するコストをそれぞれの連接規則に対して記述することができる．連接規則は，次節で述べる連接規則辞書によって定義される．

3 辞書の定義とデータ構造

3.1 辞書の概要

本システムで用いる辞書の種類について述べる．本辞書システムの特徴は，利用者によって形態品詞名や連接関係などの文法情報やを自由に定義できることである．

日本語の形態素文法は，本システムに標準的に仮定されているファイルによって定義されなければならないが，この定義はユーザに任されている．ユーザが定義すべき辞書を総称してユーザ辞書と呼ぶ．ユーザ辞書は，文法辞書と形態素辞書に大別される．文法辞書は 2 節で述べた日本語形態素文法を定義するための辞書で，形態品詞分類辞書，活用関係辞書，活用辞書，連接規則辞書から構成される (3.2 節)．形態素辞書は個々の形態素についての情報を実際に記述する辞書である (3.3 節)．

形態素解析において使用する辞書をシステム辞書と呼ぶ．システム辞書は，連接規則辞書と活用関係辞書から生成された連接対応表と連接行列，文法辞書を参照して形態素辞書に記述された情報を記号化したハッシュデータベース辞書と意味辞書から成る．

ユーザ辞書およびシステム辞書を以下にまとめる．本システムには標準的なユーザ辞書が含まれているが，ユーザはこれを自由に書き直してよい．システム辞書は，本システムによってユーザ辞書から自動的に生成されるので，ユーザはこれらを意識する必要はない．形態素解析プログラムは変換して得られたシステム辞書のみを用いる．

ユーザ辞書

辞書の種類	辞書名称 (ファイル名)	説明
文法辞書	形態品詞分類辞書 (JUMAN.grammar)	品詞分類を定義する
	活用関係辞書 (JUMAN.kankei)	活用する各品詞に対して活用型の一覧を定義する
	活用辞書 (JUMAN.katuyou)	個々の活用型の具体的な活用形の一覧を定義する
	連接規則辞書 (JUMAN.connect)	連接関係を定義する連接規則の集合
形態素辞書	各品詞辞書 (Noun.dic, etc.)	個々の形態素の定義 (3.3を参照)

システム辞書

辞書の種類	辞書名称 (ファイル名)	説明
文法辞書	連接対応表 (JUMANTREE.table)	形態品詞と接続番号との対応表
	連接行列 (JUMANTREE.matrix)	接続可能性表
形態素辞書	ハッシュデータベース辞書 (JUMANTREE.dbm.dir, JUMANTREE.dbm.pag)	辞書の各形態素が格納された ハッシュデータベース
	意味辞書 (JUMANTREE.imis)	各形態素の意味情報が格納される

本システムを使用する際に個人的にカスタマイズできる情報があり，その情報は，使用者のホームディレクトリにある.jumanrc というファイルに記述されていると仮定される．このファイルが存在し

ない場合は，本システムがもつ標準的な情報が使用される．`.jumanrc` ファイルに書かれる情報の詳細は後で詳しく述べる．

また，本システムでは標準的なユーザ辞書を用意している．これについては，3.6 節で紹介する．

文法辞書および形態素辞書ファイルでは，各行中のセミコロン ‘;’ で始まる文字列はその行末までがコメントとみなされる．

3.2 文法辞書の定義法

文法辞書の構成と内容は次のようになっている．各辞書の記述は S 式によって行なう．付録 Aに記述のフォーマットを BNF 式と例によって示す．

1. 形態品詞分類辞書： (cf. `JUMAN.grammar`)

   本システムで用いる形態品詞およびその品詞細分類の名称を定義する．一つの形態品詞およびその品詞細分類を一つの S 式 (リスト構造) によって記述する．リストの第 1 要素に形態品詞を記述し，品詞細分類があれば第 2 要素に品詞細分類名をリストによって記述する．

   個々の形態品詞と品詞細分類はリストの形で表現されるが，それに属する形態素が活用する場合にはその第 2 要素として記号 `%` を付加する．形態品詞が記号 `%` をもつ場合には，その細分類に属するすべての形態素が活用するとみなされる．

   形態品詞分類辞書の記法を付録 A.1に示す．

2. 活用関係辞書： (cf. `JUMAN.kankei`)

   活用関係辞書は，形態品詞分類辞書によって活用すると定義された形態品詞または品詞細分類それぞれに対してそれが取り得る活用型の一覧を定義する．形態品詞分類辞書で活用すると定義されたものに対しては，必ず取り得る活用型の一覧が書かれていなければならない．活用型の名前付けはユーザが自由に行なってよい．

   個々の活用関係は，二つの要素からなるリストとして定義される．第一要素は形態品詞のリストまたは形態品詞と品詞細分類からなるリストであり，活用する品詞を示す．第二要素はその品詞が取り得る活用型の一覧を示すリストであり，任意個数のアトムを含む．

   活用関係辞書の記法を付録 A.2に示す．

3. 活用辞書： (cf. `JUMAN.katuyou`)

   活用辞書は，活用関係辞書によって定義された個々の活用型に対して，それが取り得る活用形の名前およびその活用語尾を定義する．

   この辞書の各項目も二つの要素からなるリストであり，第 1 要素に活用型の名前を記述し，第 2 要素として活用形の名前と活用語尾のリストより構成されるリストを記述する．この時，活用形の名前として「基本形」を必ず記述しなければならない．後で述べる形態素辞書には，活用する形態素はその基本形が登録されていると仮定されることになっているので，個々の形態素については，形態素辞書に登録された見出し表記からこの辞書で定義される基本形の語尾を取り除いたものがその形態素の不変化部分であるとみなされる．活用語尾の存在しない活用形 (例えば，語幹) に対しては，活用語尾として $*$ と記述すればよい．

   また，不変化部分が存在しない活用語については，見出し語と基本形語尾が同じになる (例えば，付録 A.3.2のサ変動詞型の活用)．このような場合には，システムはすべての活用語尾を個別の形態素として形態素辞書に登録する．

   活用辞書の記法を付録 A.3に示す．

4. 連接規則辞書： (cf. JUMAN.connect)

   連接規則辞書は連接規則の集合である．一つの連接規則は形態素構造 (2.3節を参照) の集合の対であり，2 要素のリストによって表現される．形態素構造の集合もリストによって表現される．連接規則には第 3 要素として連接のコストを記述することができる．コストの指定がない場合のデフォルト値は “10” である．コストの値は，0 から 255 までの整数値でなければならない．ただし, 連接規則辞書から得られる連接行列では, 二つの形態素が連接不可能であることを数値 0 によって表わしているので, 連接のコストを 0 と指定することはそれらが連接不可能であることを意味する. したがって, 連接条件の例外的な記述を連接コストを 0 にすることによって行なうことができる.

   一つの連接規則の第 1 要素に含まれる形態素は第 2 要素に含まれる形態素に連接可能であることを示す．一つの形態素が複数の連接規則内の形態素構造に含まれることがあってもよい．連接関係はこれらの規則の和集合であるとみなされる．したがって，ある特定の形態素構造の対が連接可能であることが複数の連接規則によって定義されることがある．このような場合，これらの連接のコストは，連接規則辞書内で後方に現れた規則によって指定されるコストを取るとみなされる．

   形態素構造で用いられる品詞名や活用形名は文法辞書で定義されたものでなければならない．例外として，（文頭）および（文末）という形態素構造がある．前者は，文頭の直前に現れると考えられるダミーの形態品詞であり，少なくとも一つの連接規則の第 1 要素の中に現れなければならない．同様に，後者は文末の直後に繋がると考えられるダミーの形態品詞であり，少なくとも一つの連接規則の第 2 要素の中に現れる必要がある．

   連接規則辞書の記法を付録 A.4に示す．

   連接規則においては，ある品詞のグループが同じ連接関係をもつことがよく起こる．連接規則の記述を省力化するために，gcc で用意されているマクロ機能を利用することができる．JUMAN.connect.c というファイルに次のようなマクロを定義することにより，以降の連接規則内で NormalNominal という記号を #define の第二引数と等価に扱うことができる．

```
#define NormalNominal （名詞 普通名詞）\
                      （名詞 サ変名詞）\
                      （名詞 固有名詞）\
                      （名詞 地名）\
                      （名詞 人名）\
                      （名詞 数詞）\
                      （指示詞 名詞形態指示詞）\
                      （動詞 * * 基本連用形）
```

   /juman/dic に用意された trans_connect_rule というコマンドテキストを実行することにより，JUMAN.connect.c から JUMAN.connect が生成される．trans_connect_rule を実行する際には同じディレクトリ内に JUMAN.connect.c があることに注意すること．

3.3 形態素辞書の記述法

形態素辞書もリスト構造を用いて記述する．形態素辞書は，最後に ‘dic’ という拡張子をもつファイルに格納する．形態素辞書は複数のファイルに分割されていてもよい．

各形態素は，複数の見出し語を持ってもよい．また，それぞれの見出し語に対してコスト計算のための相対的な重みを指定することができる．形態素のもつコストについては，4章で説明する．複数の見出し語が記述されている場合，見出し語以外の形態素情報は共有される．

次に形態素辞書の形態素定義の記述方法を BNF で示す．

```
〈形態素定義〉          ::=  (〈#形態品詞名〉〈形態素情報の並び〉) |
                            (〈#形態品詞名〉(〈#品詞細分類名〉〈形態素情報の並び〉))
〈形態素情報の並び〉    ::=  〈形態素情報〉 | 〈形態素情報〉〈形態素情報の並び〉
〈形態素情報〉          ::=  (〈見出し語情報〉〈読み情報〉〈活用型情報〉〈意味情報〉)
〈見出し語情報〉        ::=  (見出し語 〈見出し語内容の並び〉)
〈見出し語内容の並び〉  ::=  〈見出し語内容〉 | 〈見出し語内容の並び〉
〈見出し語内容〉        ::=  〈#見出し語表記〉 | (〈#見出し語表記〉) |
                            (〈#見出し語表記〉〈#数値〉)
〈読み情報〉            ::=  (読み 〈#読み表記〉)
〈活用型情報〉          ::=  (活用型 〈#活用型名〉) | NIL
〈意味情報〉            ::=  (意味情報 〈#意味記述〉) | NIL
```

- 〈形態素情報〉は，共通の形態品詞名または品詞細分類名ごとにまとめて〈形態素情報の並び〉として書くこともできるし，個別に記述してもよい．〈#形態品詞名〉，〈#品詞細分類名〉は，「形態品詞分類辞書」で定義されていなければならない．また，品詞細分類されている形態品詞に対しては品詞細分類名を指定しなければならない．

- 〈活用型情報〉は，〈#形態品詞名〉または〈#品詞細分類名〉が活用すると定義されている時は省略できない．

- 〈#見出し語内容〉は形態素の表層の形と相対的な重みを表わす数値の対として表現する．相対的な重みが 1 の場合は，重みの記述を省略してもよい．重みとして指定できるのは正整数または小数点 1 桁以内の正小数である．活用する形態素の表層の形は，その基本形を書かねばならない．

- 〈#読み表記〉には形態素の読みを記述する．活用する形態素については，その基本形の読みを書く．

- 〈#意味記述〉には意味情報を記述する．意味情報には任意の S 式を記述することができる．また，アトムとして，二重引用符 (") で囲まれたテキストを自由に用いることができる．二重引用符に囲まれた範囲では任意の文字が使用可能である．二重引用符で囲まれたアトムの中の二重引用符は ‘\"’ によって記述可能である．記述されたデータは二重引用符を含んでそのままテキストデータとして扱われる．長さについての制限は設けていない．

付録 Bに形態素辞書の記述例を示す．

3.4　システム辞書の作成

ユーザ辞書からシステム辞書を構築するには，連接規則辞書の変換と， 2 回の辞書データ変換 (中間辞書への変換，および，システム辞書への変換) が必要である．中間辞書への変換は，変換の早い段階でユーザ辞書の構文的エラーを検出するために行なわれる．変換のためには文法辞書の情報の参照が必要である．文法辞書が存在するディレクトリの指定は，様々なオプションを記述した `.jumanrc` というファイルに記述することになっている．これについては， 3.7節で説明する．

辞書の変換に必要なプログラムは “`make all`” を実行すると，`~/juman/bin` の下に実行可能なプログラムが作られる．[3]

[3] `juman`, `juman_pl`, `makehd`, `makeint`, `makemat`, の 5 つのプログラムが生成される．

3.4.1 連接規則辞書の変換

‘makemat’ を実行することによって行なわれる．‘makemat’ は明示的な引数をもたない．この処理によって，連接規則辞書 (JUMAN.connect) および活用関係辞書 (JUMAN.kankei) から連接対応表 (JUMANTREE.table) および連接行列 (JUMANTREE.matrix) が生成される．

連接対応表は，各品詞 (形態品詞名および品詞細分類)[4]に対するエントリをもち，それから連接行列の行および列 (それぞれその品詞の後接情報および前接情報を表わす) を示すポインタを与える．活用を行なう品詞については，連接対応表のエントリにその活用形に対応するオフセットを加えた位置のエントリを参照することになる．これにより任意の形態素構造に対して一意にエントリを決定することができる．連接行列は，二つの形態素構造が連接可能かどうか，また，連接可能である場合にはその連接のコストを示す行列である．同じ後接関係を持つ形態素構造には一つの行が割り当てられる．同様に，同じ前接関係を持つ形態素構造には一つの列が割り当てられる．

3.4.2 中間辞書への変換

形態素辞書の情報はいったん中間辞書に変換される．この処理は，‘makeint’ を実行することによって行なわれる．‘makeint’ は，拡張子 ‘dic’ を持つファイルを引数に指定しなければならない．カレントディレクトリに存在するすべての形態素辞書を変換の対象にする場合には

```
makeint *.dic
```

を実行すればよい．この過程で，次の処理を行なう．

1. 形態品詞・品詞細分類・活用型・活用形を，「形態品詞分類辞書 (JUMAN.grammar)」「活用辞書 (JUMAN.katuyou)」を参照してそれぞれ 1 バイト[5]の整数値に記号化する．

2. 「連接対応表 (JUMANTREE.table)」を参照して，各形態素の連接行列へのエントリ番号を求め，4 バイトの整数値に記号化する．

3. 活用する語については，「活用辞書 (JUMAN.katuyou)」を参照して語幹のみを中間辞書に登録する．活用によって形態素の表層全体が変化するために語幹が空になるものについては，すべての活用形を展開して中間辞書に登録する．

4. ユーザ辞書の構文的な誤りを検出する．

ここで，1. と 2. の記号化を行なう理由は，それぞれの情報を固定長にしてデータ操作を容易にし，また，システム辞書ファイルの大きさを抑えるためである．記号化する前の情報は文法辞書から一意に復元できる．個々の形態素辞書ファイルに対して一つの中間辞書ファイルが生成される．中間辞書ファイルは最後の拡張子が ‘dic’ の代わりに ‘int’ になっている以外は形態素辞書ファイルと同じ名前が与えられる．

3.4.3 システム形態素辞書への変換

中間辞書からシステム形態素辞書を構築する．この処理は ‘makehd’ によって行なわれ，「ハッシュデータベース辞書」(JUMANTREE.dbm.dir, JUMANTREE.dbm.pag) と「意味辞書」(JUMANTREE.imis) が生成される．‘makehd’ には拡張子 ‘int’ をもつファイルを引数に指定しなければならない．すべての中間辞書ファイルを対象にする場合には

```
makehd *.int
```

[4]表層語が連接規則辞書に直接書かれている場合は表層語も含む

[5]このため，形態品詞・品詞細分類・活用型・活用形の分類はそれぞれ 256 通りまでしか許されない．

を実行すればよい．

makehd の実行に先だってカレントディレクトリにシステム形態素辞書が既に存在する場合には，中間辞書内の形態素がこのシステム形態素辞書に追加される．この処理においては，同じ形態品詞，品詞細分類，見出し語，読みを持つ形態素の多重登録は行なわれない．

なお，システム形態素辞書に登録された形態素を消去することも可能である．そのためには，消去を行ないたい形態素を，ユーザ形態素辞書と同様の形式で記述し，それを中間辞書に変換しておく．この中間辞書を del.int とすると

```
makehd -d del.int
```

によって，もともと del.dic にあった形態素がシステム辞書から消去される．

3.5 複数辞書の使用

JUMAN では，必要に応じて複数の辞書を使用することが出来るようになっている．ここではその使用法について述べる．

辞書は 3.1節で述べたように，文法辞書及び形態素辞書に分けられる．複数化出来るのは形態素辞書である．即ち，makehd によって作成される 3 つの辞書 (拡張子が dbm.dir, dbm.pag, imis の辞書) である．複数辞書でもこの拡張子は変わらない．また上記の一組の辞書は，末尾の拡張子以外は同じ名前をもつファイルである．

利用者が用いる形態素辞書の指定は，他のオプションと同様に，利用者のホームディレクトリにおかれた '.jumanrc' というファイルによって行なわれる．複数の辞書の指定もこのファイルに記述される．具体的な記述法については，3.7節を参照のこと．また，複数の形態素辞書を使用する場合は，それらは同じ文法辞書にしたがっていなければならない．

3.6 システム標準文法

本システムのユーザ辞書として標準の文法が用意されている．これをシステム標準文法と呼ぶ．システム標準文法を定義する文法辞書のうち，形態品詞分類辞書，活用辞書，連接規則辞書は，益岡・田窪文法 [1] を参照し，それを拡張して作成した．

1. 形態品詞分類辞書： (cf. JUMAN.grammar)

   益岡・田窪文法に「特殊」(句読点・記号・括弧など) を加え，接辞を「接頭辞」「接尾辞」に分けて，14 種類の形態品詞を定義した．

2. 活用辞書： (cf. JUMAN.katsuyou)

   益岡・田窪文法に対して，文語的表現・口語的表現・敬語表現に対応できるように拡張を行ない，21 の一般的な活用型と 7 の特殊な活用型を定義した．

3. 活用関係辞書： (cf. JUMAN.kankei)

   活用する形態構造の一覧とそれらが取り得る活用型の一覧を定義している．

4. 連接規則辞書： (cf. JUMAN.connect.c)

   新たに作成した．

3.7　オプション定義 (cf.　.jumanrc)

このファイルは形態素解析プログラムに必要な様々な選択選択肢を定義するために用いられる．オプションの定義は，‘.jumanrc’ というファイルによって行なわれる．このファイルは利用者のホームディレクトリになければならない．利用者のホームディレクトリにこの名のファイルが存在しない場合は，デフォルトのオプションファイルが参照される．デフォルトのオプションファイルの指定は，$JUMAN/Makefile[6]内の RCPATH という変数によって与える．

オプションファイルの記述例を付録 Cに示す．現在存在するオプションを次に示す．これらはすべて指定されていなければならない．

1. 文法辞書の位置
   文法辞書 (JUAMN.grammar，JUMAN.kankei，JUMAN.katuyou，JUMAN.connect，JUMANTREE.table，JUMANTREE.matrix) が存在するディレクトリを指定する．指定は次のように行なう．

   ```
   (文法ファイル /home/nltools/juman/dic)
   ```

2. 形態素辞書の位置
   形態素辞書 (.dbm.dir，.dbm.pag，.imis の拡張子をもつ一組みの辞書) が存在するディレクトリ，および，末尾の拡張子を除いた辞書ファイル名を指定する．複数組みの形態素辞書を指定することも可能である．指定は次のように行なう．

   ```
   (辞書ファイル
           /home/mydir/juman/dic/MYTREE
           /home/nltools/juman/dic/JUMANTREE )
   ```

   この例では，二組みの形態素辞書を指定している．辞書引きに際しては，これらの辞書の両方が用いられる[7]．

   使用する辞書の最大数は，5 個に設定されている．これを変更したい場合は，

   ```
   $JUMAN/lib/juman.h
   ```

   の MAX_DIC_NUMBER の値を変更しておけばよい．

3. 未定義語の品詞定義
   未定義語が発見された時にその語をどのような品詞として連接規則を適用するかを指示する．指定できるのは，文法辞書に定義されている形態素名である．品詞細分類名が存在する品詞については，品詞細分類名も定義されなければならない．次の例は，未定義語を名詞の細分類としてのサ変名詞と定義している．

   ```
   (未定義語 (名詞 サ変名詞))
   ```

4. 形態素のコストの定義
   4章で述べるように，形態素解析プログラムでは，解析結果の優先情報をコストとして計算してい

[6] $JUMAN は JUMAN システムがインストールされているディレクトリであると仮定する．以後同様．

[7] 一組みの辞書には同一の形態素の登録は行なわれないが，複数の辞書に同じ形態素が登録されている場合はあり得る．このような場合は，同じ形態素が複数得られることになる．

る．解析に曖昧性がある場合は，コストの総計が低いものを優先することになっている．文法辞書によって定義された形態素のコストを定義しなければならない．また，「未定義語」についてのコストも指定する必要がある．

品詞は，品詞名，または，品詞名と品詞細分類名の対によって表わされる．品詞細分類が定義されている品詞については、品詞細分類名 (または '*') を指定する必要がある。コストは正整数の値をとる．

```
(品詞コスト
        ((*)              10)
        ((未定義語)     5000)
        ((特殊 *)        100)
        ((動詞)          100)
        ((形容詞)        100)
        ((判定詞)         10)
        ((助動詞)         10)
        ((名詞 *)        100)
        ((名詞 固有名詞) 110)
        ((指示詞 *)      100)
        ((副詞 *)        100)
        ((助詞 *)         10)
        ((接続詞)        100)
        ((連体詞)        100)
        ((感動詞)        100)
        ((接頭辞 *)       10)
        ((接尾辞 *)       10) )
```

同じ品詞に対してコストの定義が複数回指定されている場合は，後のものが優先される．上の例では，品詞名が「名詞」の形態素のコストは 100 であるが，細分類名が「固有名詞」の形態素のコストは 110 となる．また，先頭の '(*)' の指定により，ここで明示的に定義されていない形態素のコストはすべて 10 となる．

5. 連接コストと形態素コストの相対的な重みの定義

形態素解析におけるコストの計算は形態素のコストと連接のコストの総計として計算される．これら二種類のコストに異なる重みを掛けたい場合には，それを指定することができる．解析結果のコストはそれぞれのコストにここで指定された重みを乗じた値の総計として計算される．

```
(連接コスト重み 10)
(形態素コスト重み 1)
```

また，形態素解析の過程において，常にコストが最低の結果を出すのではなく，ある程度のコスト幅を許容したい場合がある．この許容幅を指定することができる．

```
(コスト幅 200)
```

コスト計算については， 4章を参照のこと．

4 形態素解析

形態素解析に必要な文法や木構造辞書の存在場所，並びに，解析中で用いられるコストに関する基本的な情報は前出の `.jumanrc` に記述されている．

JUMAN は，EUC コードの日本語文字列を入力とし，連接規則辞書によって許容された形態素からなる束 (lattice) 状の構造を出力とする．ただし，結果の表示については，解に含まれる形態素情報を 1 形態素毎に 1 行に表示するだけである．表示についてはいくつかのオプションが用意されている．

4.1 形態素解析アルゴリズム

JUMAN の解析アルゴリズムは，入力としてあたえられた日本語の文字列に対する次の基本動作よりなる．改行をもって一つの入力文字列の終了とする．

- ある特定の位置からはじまるすべての可能な形態素を辞書引きによって得る．
- 辞書引きによって得られた個々の形態素に対して，その直前の位置に存在するすべての形態素との連接可能性のチェック，および，コストの計算を行なう．
  - 連接可能性のチェックによって連接不可能とわかった形態素間の接続は行なわれない．また，その位置で連接可能なもののうち最良 (最小) のコストと比較して `.jumanrc` の（コスト幅 ??）によって定義される数値以上ののコスト差をもつ形態素の接続は行なわれない．

本形態素解析では次の二種類のコストが使用される．

形態素のコスト: 個々の形態素に与えられているコストである．形態素のコストは，`.jumanrc` で定義された「品詞コスト」，「形態素コスト重み」，および，形態素辞書で定義された見出し語のコストに対する重み，の三つの数値の積として計算される．

連接コスト: 二つの形態素の連接のコストである．連接規則辞書で定義されたコストと `.jumanrc` で定義された「連接コスト重み」の積として計算される．

形態素解析の一つの解析結果は形態素からなる列であり，その総コストは，それに含まれる形態素のコストの総計，および，各形態素間の連接コストの総計の和である．ただし，列の先頭と末尾には，それぞれ，「文頭」および「文末」と呼ばれるダミーの形態素があると仮定する．

一般に解析結果は，文頭および文末を両端点とする束状のグラフ構造になる．このグラフ内の文頭から文末へいたるそれぞれの経路が一つの可能な解析結果を表わすことになる．

連接可能性のチェックおよびコストの計算の詳細について説明する．最初の辞書引きは文字列の先頭で行なわれ，その直前には「文頭」が存在すると仮定される．また，「文頭」がもつコストは 0 である．

最初に辞書引きが行なわれるのは入力文字列の先頭 (すなわち，「文頭」の直後) である．以降，辞書引きが行なわれるのは入力文字列中において文頭からの有効な連接をもつ形態素の直後の位置である．形態素の有効性，無効性については，以下の説明の中で述べる．

ある位置で辞書引きによって得られるすべての形態素を $Mf_j(1 \le j \le n)$ とし，その位置の直前で終了する形態素が $Mb_i(1 \le i \le m)$ であるとする．各 Mb_i は文頭から自分自身へいたる束状のグラフの経路の中で最小のコストの値を保持している．そのコストを Cb_i とする．

それぞれの Mf_j について，すべての Mb_i との連接可能性のチェックを連接行列により行なう．Mb_i の後接情報と Mf_j の前接情報により，それらが連接可能であるか，また，連接可能であればその連接コ

ストを求める．Mb_i と Mf_j の連接コストを $C_{i,j}$ とする (連接不可能な場合 $C_{i,j} = \infty$ とする)．したがって，Mf_j に対するすべての連接コストは，

$$Cb_1 + C_{1,j},\ Cb_2 + C_{2,j},\ \cdots, Cb_m + C_{m,j}$$

となる．これらの最小値からの差が `.jumanrc` の「コスト幅」以内であるものだけが Mf_j に連接可能であると考えられる．連接可能な Mb_i が存在しない場合 (すべての $C_{i,j}$ が無限大の場合)，Mf_j は無効な形態素と考えられ，以後の解析結果のグラフには含まれない．Mf_j が無効でないとき，Mf_j は最小の $Cb_i + C_{i,j}$ の値に自身の形態素コストを加えた値を解析結果のコストとして保持することになる．

形態素解析の最後の段階では，入力文字列の末尾に現れる有効な形態素と「文末」との連接可能性のチェックとコスト計算が行なわれ，解析結果としてのグラフ構造が得られる．

4.2 未定義語の取り扱い

未定義語に関しては，本システムでは，入力文字列中のあらゆる位置から未定義語が存在する可能性を考慮している．ひらかなおよび漢字以外の文字については，同種の文字の終りまで (カタカナ，アルファベット，数字 等) をまとまった語として一つの未定義語と解釈する．ひらかなと漢字については，一文字ずつを未定義語の候補と考えている．

切り出された未定義語に対する連接関係を定義するために，未定義語をどのような品詞と考えるかは，オプション定義ファイル (`.jumanrc`) 内の「未定義語品詞」によって自由に指定できる．ただし，未定義語品詞は，文法辞書で定義された品詞名 (品詞再分類がある場合には品詞再分類名も指定する必要がある) のいずれか一つでなければならない．

未定義語に対するコストは同じく`.jumanrc` の「品詞コスト」欄で定義できるので，これに高いコストを与えることにより，未定義語を含む解析結果の優先度を下げることができる．

4.3 C 版

JUMAN システムのインストールについては，付録 Dを参照のこと．システムの実行ファイルは，`$JUMAN/bin/juman` である．形態素解析の実行については，いくつかのオプションが用意されている．以下にそれをまとめる．

- 形態素解析の実行

  ```
  juman -[b|m|p] -[f|e|c]
  ```

  によって起動され，標準入力から一行ごとに文を読み込んで処理を行なう．

- 処理内容
  コスト最小 (それぞれの形態素の区切りで最小コストとの差が許容されるコスト幅以内) の解を求め，結果をオプションに従って表示する．各オプションの意味は次の通りである．

  – 解が曖昧性を含む場合の表示方法 (曖昧性がない場合はどの方法も同じ表示となる)

    `-b` 後方最長一致の解を一つだけ表示する (default)
    `-m` 曖昧性のある部分だけ，複数の形態素を表示する
    `-p` 曖昧性の組合せを展開し，すべての解を個別に表示する

  – 各形態素の表示方法

-f　カラムを整えて表示 (default)
-e　完全な形態素情報を文字で表示
-c　完全な形態素情報をコードで表示

juman はフィルターとしても使用できる．以下に使用例を示す．

使用例

```
% juman
私は昨日学校へ行った．
私      (わたくし)    私    普通名詞
は      (は)          は    副助詞
昨日    (きのう)      昨日  普通名詞
学校    (がっこう)    学校  普通名詞
へ      (へ)          へ    格助詞
行った  (おこなった)  行う  動詞      子音動詞ワ行  タ形
．      (．)          ．    句点
EOS
```

```
% cat > temp
私は昨日学校へ行った．
nltools% cat temp | juman -m
私      (わたくし)    私    普通名詞
私      (わたし)      私    普通名詞
は      (は)          は    副助詞
昨日    (きのう)      昨日  普通名詞
昨日    (さくじつ)    昨日  普通名詞
学校    (がっこう)    学校  普通名詞
へ      (へ)          へ    格助詞
行った  (おこなった)  行う  動詞      子音動詞ワ行    タ形
行った  (いった)      行く  動詞      子音動詞カ行促  タ形
．      (．)          ．    句点
EOS
```

4.4 Prolog 版

- 基本的構成
  C 版のシステムが形態素解析を行なった結果を，SICStus Prolog がソケットを介して受けとるという構成になっている．具体的には，C 側でメイン・サーバーが常駐しており，SICStus Prolog 側から要求があれば子プロセスが一つユーザに割り当てられ，その子プロセス内で C 版の形態素解析プログラムが起動される．

- 動作環境
  JUMAN が仮定する Prolog 処理系は，SICStus Prolog 2.1 である．日本語を用いるためには，環境変数 SP_CTYPE の値を euc に設定する必要がある．具体的には，例えば次の一行を例えば利用者の.login ファイルに含めればよい．

```
setenv SP_CTYPE euc
```

- 起動方法

  Prolog を起動し，‘`$JUMAN/juman_pl/juman_user.pl`’ を consult する．

  ‘`$JUMAN/juman_pl/juman_user.pl`’ はユーザ設定用ファイルなので，ユーザが各自のディレクトリにコピーして各自の設定を行なうことを前提としている．

  次に，`juman_start_server` を実行して，C 側のメイン・サーバーを起動する．既にメイン・サーバーが起動されている場合は，新たに起動されない．

  解析は，以下の要領でファイル単位で行なうことができる．

  ```
  | ?- juman.
  Input file name?  入力ファイル名.
  Output file name?  出力ファイル名.
  ```

  入力ファイル名，出力ファイル名を “user” とすると，端末からの入出力が可能である．C 側では，一回の `juman` の呼び出しによって子プロセスが起動されて，`juman` の呼び出しが終了すると子プロセスも終了する．

  また，Prolog 側からメイン・サーバーを終了させるには，`juman_kill_server` を実行する．

- 処理内容

  Prolog 版は，C 版のシステムが形態素解析を行なった結果を受けとるだけなので，形態素解析の処理内容は C 版と全く同じである．一つの形態素について，Prolog 版が C 版から受けとるデータは，

  morph([識別子 (ID), 開始位置 (From), 終了位置 (To), スコア (Score),
  見出し語 (Md), 読み (Yd), 基本形 (Kh), 品詞名 (Hd), 活用型名 (KT),
  活用形名 (KF), 意味情報 (Im), 前接形態素の識別子のリスト (PreIDL)])

  という複合項である．また，C 版からデータを受けとる処理は，‘`$JUMAN/juman_pl/juman.pl`’ 中の`juman/3` において，`read(+`*Str*,-*MorphList*`)` によって上述の複合項のリスト *MorphList* を読み込むという形で行なわれている．

  Prolog 版には出力オプションはない．基本的には C 版の ‘`juman -m -e`’ と同様の出力結果を返す．

- ソケット通信について

  一般にソケット通信を行なう際には，ホスト名とポート番号を指定する必要がある．

  Prolog 版 JUMAN においては，C 側のメイン・サーバーを起動するホスト及び使用するポート番号を自由に設定できる．この設定は，‘`$JUMAN/juman_pl/juman_user.pl`’ 中の`juman_server_host/1`，`juman_server_port/1` の引数を変更することによって行なう．ポート番号のデフォルトは，16010 である．

参考文献

[1] 益岡隆志，田窪行則：『基礎日本語文法』くろしお出版，1989．

[2] 妙木裕，松本裕治，長尾眞：「汎用日本語辞書および形態素解析システム」情報処理学会第 42 回全国大会予稿集，1991．

[3] 中村俊久，黒橋禎夫，長尾真 「部分文字列情報の利用による日本語単語の高速検索」 情報処理学会自然言語処理研究会 NL-101, 1994．

付録

A　ユーザ辞書の記法の定義と例

ユーザ辞書の記法を BNF によって示す．以下，〈···〉は非終端記号名を表わす．〈#形態品詞名〉のように # が付けられた非終端記号はユーザによって定義される具体的な名前を表わすと仮定する．NIL は空リストではなく，空系列を表わす．

A.1　形態品詞分類辞書 (JUMAN.grammar) の記述

A.1.1　形態品詞分類辞書の個々の項目の記述

```
〈形態品詞分類辞書項目〉   ::=  (〈形態品詞〉) | (〈形態品詞〉 (〈品詞細分類の並び〉))

〈形態品詞〉             ::=  (〈#形態品詞名〉) | (〈#形態品詞名〉 %)
〈品詞細分類の並び〉      ::=  〈品詞細分類〉〈品詞細分類の並び〉
〈品詞細分類〉           ::=  (〈#品詞細分類名〉) | (〈#品詞細分類名〉 %)
```

A.1.2　形態品詞分類辞書の定義例

```
((動詞 %))                          ; 動詞は形態品詞名であり，活用する．

((名詞)                             ; 名詞は形態品詞であり，
        ((普通名詞)                  ; 普通名詞，サ変名詞，固有名詞 ... に
         (サ変名詞)                  ; 細分類される．
         (固有名詞)
         (数詞)
         (形式名詞)
         (副詞的名詞)))

((接尾辞)
        ((名詞性述語接尾辞)
         (名詞性名詞接尾辞)
         (名詞性名詞助数辞)
         (形容詞性述語接尾辞 %)     ; 形容詞性述語接尾辞は活用する．
         (形容詞性名詞接尾辞 %)
         (動詞性接尾辞 %)))
```

A.2 活用関係辞書 (JUMAN.kankei) の記述

A.2.1 活用関係辞書の個々の項目の記述

```
〈活用関係辞書項目〉   ::=  (〈品詞〉(〈活用型の並び〉))

〈品詞〉               ::=  (〈#形態品詞名〉) | (〈#形態品詞名〉〈#品詞細分類名〉)
〈活用型の並び〉       ::=  〈#活用型名〉 | 〈#活用型名〉〈活用型の並び〉
```

A.2.2 活用関係辞書の定義例

```
((形容詞)                          ; 形容詞はイ形容詞とナ形容詞という活用型をもつ
          (イ形容詞
           ナ形容詞))

((助動詞)
         (イ形容詞
          ナ形容詞
          判定詞
          無活用型
          助動詞ぬ型
          助動詞だろう型
          助動詞そうだ型))

((接尾辞 形容詞性述語接尾辞)   ; 接尾辞の中で形容詞性述語接尾辞に細分類される
         (ナ形容詞                 ; 形態素はナ形容詞またはイ形容詞という活用型を
          イ形容詞))               ; もつ
```

A.3 活用辞書 (JUMAN.katuyou) の記述

A.3.1 活用辞書の個々の項目の記述

```
〈活用辞書項目〉     ::=  (〈#活用型名〉(〈活用形対の並び〉))

〈活用形対の並び〉   ::=  〈活用形対〉| 〈活用形対〉〈活用形対の並び〉
〈活用形対〉         ::=  (〈#活用形名〉〈語尾表示〉)
〈語尾表示〉         ::=  〈#語尾〉 | *
```

A.3.2 活用辞書の定義例

```
(母音動詞
    ((語幹          *   )
     (基本形        る  )
     (未然形        *   )
     (意志形        よう)
     (命令形        ろ  )
     (命令形        よ  )
     (基本条件形    れば)
     (基本連用形    *   )
     (タ形          た  )
     (タ系条件形    たら)
     (タ系連用テ形  て  )
     (タ系連用タリ形 たり))
)

(サ変動詞
    ((基本形        する  )
     (未然形        さ    )
     (意志形        しよう)
     (命令形        しろ  )
     (命令形        せよ  )
     (基本条件形    すれば)
     (基本連用形    し    )
     (タ形          した  )
     (タ系条件形    したら)
     (タ系連用テ形  して  )
     (タ系連用タリ形 したり))
)
```

A.4　連接規則辞書 (JUMAN.connect) の記述

A.4.1　連接規則の記述

```
〈連接規則〉          ::=  ( (〈形態素構造の並び〉) (〈形態素構造の並び〉) )  |
                          ( (〈形態素構造の並び〉) (〈形態素構造の並び〉) 〈# 連接コスト〉)
〈形態素構造の並び〉  ::=  〈形態素構造〉 | 〈形態素構造〉〈形態素構造の並び〉
〈形態素構造〉        ::=  (〈# 形態品詞名〉)  | (〈# 形態品詞名〉〈# 品詞細分類名〉)  |
                      (〈# 形態品詞名〉〈# 品詞細分類名〉〈# 活用型名〉)  |
                      (〈# 形態品詞名〉〈# 品詞細分類名〉〈# 活用型名〉〈# 活用形名〉)  |
                      (〈# 形態品詞名〉〈# 品詞細分類名〉〈# 活用型名〉〈# 活用形名〉〈# 見出し語〉)
```

〈# 形態品詞名〉〈# 品詞細分類名〉〈# 活用型名〉〈# 活用形名〉〈# 見出し語〉については，＊も指定できる．
〈# 連接コスト〉は 0 から 255 までの整数値 (省略された場合はのデフォルト値は 10)．

A.4.2　連接規則の定義例

```
(((名詞)                              ; 名詞，指示詞の名詞形態指示詞などは
  (指示詞 名詞形態指示詞)             ; 判定詞および特殊の読点と連接可能
  (接尾辞 名詞性述語接尾辞)
  (接尾辞 名詞性名詞接尾辞)
  (接尾辞 名詞性名詞助数辞))
        ((判定詞)
         (特殊 読点))
)

(((名詞 サ変名詞)                         ; サ変名詞と接尾辞の「化」には，
  (接尾辞 名詞性名詞接尾辞 * * 化))       ; 「する」「できる」などが
        ((動詞 * サ変動詞 * する)         ; 後接可能である．
         (動詞 * 母音動詞 * できる)       ; また，その連接のコストは 5 であり
         (動詞 * 母音動詞 * 出来る))      ; 結合度が強い．
        5 )
```

B　形態素辞書の記述

B.1　形態素の記述

```
〈形態素定義〉          ::=  (〈#形態品詞名〉〈形態素情報の並び〉) |
                          (〈#形態品詞名〉(〈#品詞細分類名〉〈形態素情報の並び〉))
〈形態素情報の並び〉    ::=  〈形態素情報〉 | 〈形態素情報〉〈形態素情報の並び〉
〈形態素情報〉          ::=  (〈見出し語情報〉〈読み情報〉〈活用型情報〉〈意味情報〉)
〈見出し語情報〉        ::=  (見出し語 〈見出し語内容の並び〉)
〈見出し語内容の並び〉  ::=  〈見出し語内容〉 | 〈見出し語内容の並び〉
〈見出し語内容〉        ::=  〈#見出し語表記〉 | (〈#見出し語表記〉) |
                          (〈#見出し語表記〉〈#数値〉)
〈読み情報〉            ::=  (読み 〈#読み表記〉)
〈活用型情報〉          ::=  (活用型 〈#活用型名〉) | NIL
〈意味情報〉            ::=  (意味情報 〈#意味記述〉) | NIL
```

〈活用型情報〉については，活用すると定義された形態素に対しては省略することができない．
〈意味情報〉は，省略可能．

B.2　形態素の記述例

```
(名詞
  (普通名詞
    ((見出し語 日本語)
     (読み にほんご))
    ((見出し語 英語)
     (読み えいご))
  )
)

(動詞
    ((見出し語 (歩く) (あるく 1.5))
     (読み あるく)
     (活用型 子音動詞カ行)
     (意味情報 "walk(X)")
    )
)
```

C　オプションファイル (.jumanrc) の記述例

```
(文法ファイル /home/nltools/juman/dic)

(辞書ファイル /home/nltools/juman/dic/JUMANTREE)

(未定義語品詞 (名詞 サ変名詞))

(品詞コスト
        ((*)             10)
        ((未定義語)    5000)
        ((特殊 *)       100)
        ((動詞)         100)
        ((形容詞)       100)
        ((判定詞)        10)
        ((助動詞)        10)
        ((名詞 *)       100)
        ((指示詞 *)     100)
        ((副詞 *)       100)
        ((助詞 *)        10)
        ((接続詞)       100)
        ((連体詞)       100)
        ((感動詞)       100)
        ((接頭辞 *)      10)
        ((接尾辞 *)      10) )

(連接コスト重み 10)

(形態素コスト重み 1)

(コスト幅 0)
```

D　インストレーション手順

JUMAN システムが存在するディレクトリを `$JUMAN` とする．システムをインストールする際には，使用者のホームディレクトリにある `.jumanrc` 内の「文法ファイル」に，文法辞書が存在するディレクトリを指定しておくこと．なお，使用者のホームディレクトリに `.jumanrc` ファイルが存在しない場合には，3.7節で述べたように，デフォルトのオプション定義ファイルが参照される．

D.1　システムの生成

`$JUMAN` ディレクトリで `make all` を実行する．これによって，juman システムが作られる．

D.2　システム辞書の生成

`$JUMAN/dic` ディレクトリで以下のコマンドを実行する．

1. `trans_connect_rule`
   これにより，`JUMAN.connect.c` が変換されて，連接規則辞書 `JUMAN.connect` が得られる．既に `JUMAN.connect` が存在する場合や，`JUMAN.connect.c` を変更した場合以外は，これを実行する必要はない．

2. `$JUMAN/bin/makemat`
   これにより連接対応表 (`JUMANTREE.table`) と連接行列 (`JUMANTREE.matrix`) が生成される．実行中，最初に表示される数値は連接規則辞書に含まれる異なる形態素構造ののべ総数を表わす．最後に表示される二つの数値は，同じ接続属性をもつ形態素構造を縮退して最終的に得られる連接行列の行と列の大きさである．

   (注： makemat.c の先頭に，一つの活用型に対する活用形の種類の上限，連接対応表，および，連接行列を作るための中間行列，のサイズの上限が指定されている．大きな接続規則辞書に対してはこれを大きくする必要がある．)

3. `$JUMAN/bin/makeint *.dic`
   これにより，辞書ファイルが中間ファイル (`*.int`) に変換される．

4. `rm JUMANTREE.dbm.*`
   ハッシュデータベース辞書を消去する．初めて木構造辞書を作る場合，もしくは，辞書へ単語を追加する場合は，これを実行する必要はない．

5. `$JUMAN/bin/makehd *.int`

   システム標準辞書を変換してシステム辞書が生成される．変換に要する時間は，辞書の大きさとマシンの性能にもよるが，現在の辞書では，SPARCstation10/30 上で 15 分程度である．辞書のコンパイルにはファイルアクセスが頻繁に起こるため，なるべく辞書をローカルに持っているマシンの上でコンパイルすることが好ましい．現在の標準辞書により最終的に生成されるファイルは以下の通り．

`JUMANTREE.dbm.dir`	8192	バイト
`JUMANTREE.dbm.pag`	63455232	バイト
`JUMANTREE.imis`	0	バイト

E　JUMAN 0.8 から JUMAN 1.0 への拡張点

1. ICOT 辞書との結合により語彙総数が大幅に増加した (異なり形態素数約 13 万語)
2. 複数の形態素辞書を同時に使用できるようにした．
3. MCC 自然言語処理研究グループにより辞書引きの高速化のためのハッシング機能が追加された．
4. 解析中に使用するコストの計算方法を統一的なものにした．
   - 形態素自身にコストを持たせた．
   - 連接コストを連接規則個別に定義できるようにした．
   - 形態素コストと連接コストに重みを付けることができるようにした．
   - 形態素間の連接点で許容可能なコストの幅を指定できるようにした．
5. 複数の見出し語 (かな表記を含む) をもつ形態素を単一の形態素として記述できるようした．また，それぞれの見出し語に相対的な重みを記述できるようにした．
6. 連接規則辞書中でマクロ定義を行なえるようにした．
7. 個人ユーザが記述するオプショナルな情報を `.jumanrc` というファイルにまとめて記述するように変更した．
8. C 版のシステムからソケットを介して結果が SICStus Prolog に渡るようになった．これに伴い，Prolog 版の形態素解析プログラムが不要になった．
9. システム標準文法における助詞の分類のうち,「提題助詞」と「取り立て助詞」の区別を止め,「副助詞」に統一した.

F　JUMAN 1.0 から JUMAN 2.0 への拡張点

1. UNIX に標準装備のハッシュデータベース ndbm を用い，さらに，疑似的な TRIE 構造を実現することにより，辞書引きを高速化した．
2. システムのコンパイルを従来の `gmake` ではなく，`make` により実行できるようにした．