

日本語形態素解析システム
JUMAN version 5.1

黒橋 禎夫 河原 大輔

平成 17 年 9 月

Japanese Morphological Analysis System JUMAN 5.1 Users Manual

Copyright (c) 2005 University of Tokyo

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name University of Tokyo may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY University of Tokyo “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE University of Tokyo BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Version 0.6	17 February 1992
Version 0.8	14 April 1992
Version 1.0	25 February 1993
Version 2.0	11 July 1994
Version 3.0b	11 June 1996
Version 3.0	16 October 1996
Version 3.1	15 November 1996
Version 3.11	5 December 1996
Version 3.2	22 May 1997
Version 3.3	5 September 1997
Version 3.4	11 November 1997
Version 3.5	23 March 1998
Version 3.6	2 December 1998
Version 3.61	26 May 1999
Version 4.0	19 July 2003
Version 5.0	23 March 2005
Version 5.1	6 September 2005

目次

1	はじめに	1
2	日本語形態素文法	3
2.1	形態品詞および品詞細分類	3
2.2	活用型と活用形	3
2.3	形態素構造	3
2.4	接続規則と接続関係	4
3	辞書の定義とデータ構造	4
3.1	辞書の概要	4
3.2	文法辞書の定義法	5
3.3	形態素辞書の記述法	7
3.4	複数辞書の使用	8
3.5	システム標準文法	8
3.6	リソースファイル (jumanrc)	8
4	形態素解析	10
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
5	インストール方法	12
5.1	Unix 系 OS におけるインストール方法	12
5.2	Windows におけるインストール方法	13
6	JUMAN の使用法	13
6.1	スタンドアロンモードでの使用	13
6.2	サーバ・クライアントモードでの使用	15
6.2.1	サーバの起動	15
6.2.2	クライアントの起動	15
6.3	Perl からの利用	16
	参考文献	16
	付録	17
A	ユーザ辞書の記法の定義と例	17
A.1	形態品詞分類辞書 (JUMAN.grammar) の記述	17
A.1.1	形態品詞分類辞書の個々の項目の記述	17
A.1.2	形態品詞分類辞書の定義例	17
A.2	活用関係辞書 (JUMAN.kankei) の記述	18
A.2.1	活用関係辞書の個々の項目の記述	18
A.2.2	活用関係辞書の定義例	18
A.3	活用辞書 (JUMAN.katuyou) の記述	19

A.3.1	活用辞書の個々の項目の記述	19
A.3.2	活用辞書の定義例	19
A.4	接続規則辞書 (JUMAN.connect) の記述	20
A.4.1	接続規則の記述	20
A.4.2	接続規則の定義例	20
B	形態素辞書の記述	21
B.1	形態素の記述	21
B.2	形態素の記述例	21
C	リソースファイル (jumanrc) の記述例	22
D	JUMAN サーバのプロトコルの概要	23
E	JUMAN の更新履歴	23
E.1	JUMAN 0.8 から JUMAN 1.0 への拡張点	23
E.2	JUMAN 1.0 から JUMAN 2.0 への拡張点	24
E.3	JUMAN 2.0 から JUMAN 3.0 への拡張点	24
E.3.1	連語処理	25
E.3.2	数詞の連結処理	25
E.3.3	括弧, 空白の透過処理	26
E.3.4	カタカナ語の扱い	26
E.3.5	解析結果の表示オプションの追加	26
E.3.6	標準辞書の整備	26
E.3.7	標準文法の整備	27
E.3.8	辞書システムの GDBM バージョンの追加	28
E.4	JUMAN 3.0 から JUMAN 3.1 への拡張点	28
E.5	JUMAN 3.1 から JUMAN 3.2 への拡張点	28
E.6	JUMAN 3.2 から JUMAN 3.3 への拡張点	29
E.7	JUMAN 3.3 から JUMAN 3.4 への拡張点	29
E.8	JUMAN 3.4 から JUMAN 3.5 への拡張点	29
E.9	JUMAN 3.5 から JUMAN 3.6 への拡張点	30
E.10	JUMAN 3.6 から JUMAN 3.61 への拡張点	30
E.11	JUMAN 3.61 から JUMAN 4.0 への拡張点	30
E.12	JUMAN 4.0 から JUMAN 5.0 への拡張点	30
E.12.1	見出し語の整備	31
E.12.2	自立語辞書の基本 format の整理	32
E.12.3	表記バリエーションの整理	33
E.12.4	読みの音訓の情報の付与	35
E.12.5	「読ます」の形の使役「読ませる」の形の使役受身への対応	35
E.13	JUMAN 5.0 から JUMAN 5.1 への拡張点	36

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が次の2点です。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。

本システムの辞書は、JUMAN2.0までのバージョンでは、長尾研究室で行われた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しました。また、JUMAN3.0以降のバージョンでは、(株)日本電子化辞書研究所(EDR)から許諾を得てEDR日本語単語辞書の一部を利用しています。

本システムの辞書は、JUMAN2.0までのバージョンでは、長尾研究室で行われた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しました。また、JUMAN3.Xでは、(株)日本電子化辞書研究所(EDR)から許諾を得てEDR日本語単語辞書の一部を利用しました。また、JUMAN4.0以降のバージョンでは独自に構築した辞書を利用しています。

本システムの開発は京都大学長尾研究室を中心に、他の多くの機関の方々の協力を得て行われました。長尾研究室の松本裕治先生(現 奈良先端大)には常に開発の指導的役割をはたして頂きました。妙木裕さん(現 キヤノン)にはJUMANの初期バージョンを開発していただきました。米国MCCの自然言語グループの方々にはJUMAN1.0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健伸先生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JUMAN2.0で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。長尾研究室の山地治さんには連語処理をはじめとしたJUMAN3.0へのバージョンアップをしていただきました。NTT基礎研の磯崎秀樹さんにはJUMAN3.4へのバージョンアップに伴うSICStus Prologとの接合部の調整をしていただきました。また、JUMAN3.1以降では奈良先端大松本研究室で開発された茶笥のパトリシア木による辞書検索システムを利用させていただいています。また、JUMAN3.5でのサーバ・クライアント方式への拡張は同じく茶笥を参考にさせていただき、その拡張作業には学術情報センターの影浦峽先生、吉岡真治先生に多大なご協力頂きました。言語メディア研(旧長尾研)の工藤拓さんにはWindows版の作成とPerlモジュール化をしていただきました。また、一人一人お名前を挙げることはできませんが、JUMANシステムに対して多くの方々からコメントと質問をいただきました。御協力頂いた各位にお礼申し上げます。

平成 17 年 8 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。

〒113-8656 東京都文京区本郷 7-3-1

東京大学大学院 情報理工学系研究科 電子情報学専攻

黒橋 禎夫

Tel: (03)5841-6672

Fax: (03)5841-6689

Email: nl-resource@kc.t.u-tokyo.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、

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

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。

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接続規則を記述する際に使われる。

2.4 接続規則と接続関係

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、それらは接続可能であるといい、それを記述した規則を接続規則と呼ぶ。また、接続規則によって定義される 2 項関係を接続関係と呼ぶ。

接続規則は二種類の形態素の接続可能性が記述可能である。また、接続の強弱に関するコストをそれぞれの接続規則に対して記述することができる。接続規則は、次節で述べる接続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接続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、接続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、接続規則辞書と活用関係辞書から生成された接続対応表と接続行列、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データベース辞書とインデックスファイルからなる。

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
	接続規則辞書 (JUMAN.connect.c)	接続関係を定義する接続規則の集合
形態素辞書	ContentW.dic など	個々の形態素の定義 (3.3 を参照)

システム辞書

²この例のように、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。

辞書の種類	辞書名称 (ファイル名)	説明
文法辞書	接続対応表 (jumandic.tab) 接続行列 (jumandic.mat)	形態品詞と接続番号との対応表 接続可能性表
形態素辞書	データベース辞書 (jumandic.dat) インデックスファイル (jumandic.pat)	各形態素の情報が格納される パトリシア木によるインデックス

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` というファイルで設定できる。このファイルが存在しない場合は、システムの標準的な設定ファイルが使用される。設定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、3.5 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。

バージョン 3.4 以降では未定義語も一つの品詞として扱い、その細分類として「カタカナ」、「アルファベット」、「その他」をたてている。これらは形態品詞分類辞書で必ず定義する必要がある (4.2 節参照)。

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)

(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)

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，その基本形を書かなければならない．
- #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 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，データベース辞書 (jumandic.dat) ・ インデックスファイル (jumandic.pat) からなる形態素辞書である．

利用者が用いる形態素辞書の指定は，他の設定と同様にリソースファイルによって行われる．具体的な記述法については，3.6 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．

3.5 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.katuyou)

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．

3.6 リソースファイル (jumanrc)

形態素解析プログラムに必要な様々な設定はリソースファイルによって指定する．ユーザのホームディレクトリに「.jumanrc」というファイルがあればこれがリソースファイルとして用いられる．ユーザ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，\$PREFIX/etc/jumanrc が参照される．

以下に必要な設定項目を示す．これらはすべて指定されていなければならない．(リソースファイルの記述例を付録 C に示す．)

1. 文法辞書の位置

文法辞書 (JUAMN.grammar, JUMAN.kankei, JUMAN.katuyou, JUMAN.connect, jumandic.tab, jumandic.mat) 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う．

(文法ファイル /usr/local/share/juman/dic)

2. 形態素辞書の位置

形態素辞書 (jumandic.dat, jumandic.pat が存在するディレクトリを指定する．複数の形態素辞書ディレクトリを指定することも可能である．指定は次のように行う．

(辞書ファイル /home/mydir/juman/dic
/usr/local/share/juman/dic)

この例では，二つ形態素辞書ディレクトリを指定している．辞書引きに際しては，これら両方のディレクトリ中の形態素辞書が用いられる³．

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，配布パッケージの lib/pat.h の MAX_DIC_NUMBER の値を変更してからインストールを行えばよい．

3. 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．

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)	
((特殊 *)	100)	
((動詞)	100)	
((形容詞)	100)	
((判定詞)	11)	
((助動詞)	10)	
((名詞 *)	100)	
((名詞 数詞)	40)	
((名詞 形式名詞)	70)	
((名詞 副詞的名詞)	70)	
((指示詞 *)	40)	
((指示詞 副詞形態指示詞)	60)	
((副詞 *)	100)	
((助詞 *)	10)	
((助詞 終助詞)	20)	

³一つの形態素辞書には同一の形態素の登録は行われないが，複数の形態素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．

((接続詞)	100)
((連体詞)	100)
((感動詞)	100)
((接頭辞 *)	50)
((接尾辞 名詞性述語接尾辞)	14)
((接尾辞 名詞性名詞接尾辞)	35)
((接尾辞 名詞性名詞助数辞)	35)
((接尾辞 名詞性特殊接尾辞)	35)
((接尾辞 形容詞性述語接尾辞)	14)
((接尾辞 形容詞性名詞接尾辞)	14)
((接尾辞 動詞性接尾辞)	14)
((未定義語 カタカナ)	5000)
((未定義語 アルファベット)	100)
((未定義語 その他)	5000)

)

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「数詞」の形態素のコストは 40 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。

4. 接続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接続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。

(接続コスト重み 4)

(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 0)

コスト計算については、4 章を参照のこと。

* 未定義語の品詞定義

バージョン 3.4 以降ではリソースファイルでの未定義語の品詞定義は不必要となった。

4 形態素解析

JUMAN は、EUC コードの日本語文字列を標準入力から一行ごとに読み込んで入力とし、接続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。

4.1 形態素解析アルゴリズム

JUMAN の解析アルゴリズムは、入力として与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始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接続可能性のチェック、および、コストの計算を行う。
 - 接続可能性のチェックによって接続不可能と分かった形態素間の接続は行われない。また、その位置で接続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コスト差をもつ形態素の接続は行われない。

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。

接続コスト：二つの形態素の接続のコストである。接続規則辞書で定義されたコストと jumanrc で定義された「接続コスト重み」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、各形態素間の接続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。

接続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、辞書引きが行われるのは入力文字列中において文頭からの有効な接続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 $Mf_j (1 \leq j \leq n)$ とし、その位置の直前で終了する形態素が $Mb_i (1 \leq i \leq m)$ 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 との接続可能性のチェックを連接行列により行う。 Mb_i の後接情報と Mf_j の前接情報により、それらが接続可能であるか、また、接続可能であればその接続コストを求める。 Mb_i と Mf_j の接続コストを $C_{i,j}$ とする (接続不可能な場合 $C_{i,j} = \infty$ とする)。したがって、 Mf_j に対するすべての接続コストは、

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

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 に接続可能であると考えられる。接続可能な 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」との接続可能性のチェックとコスト計算が行われ, 解析結果としてのグラフ構造が得られる。

出力としてはコスト最小 (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 (オプションについては 6 節)。

4.2 未定義語の取り扱い

本システムでは, 入力文字列中のあらゆる位置で未定義語が存在する可能性を考慮している。平仮名および漢字については一文字ずつを一語の未定義語としてきりだす。それ以外の文字については, 同種の文字 (カタカナ, アルファベット, 数字 等) の終りまでをまとめた一語の未定義語とする。そして, カタカナ文字列には「カタカナ」, アルファベット文字列には「アルファベット」, それ以外には「その他」という品詞細分類を与える。

未定義語という品詞と, その細分類である「カタカナ」「アルファベット」「その他」は形態品詞辞書で定義しておかなければならない。接続関係を接続規則辞書で定義すること, コストをリソースファイルの「品詞コスト」欄で定義することは通常の品詞と同様である。

5 インストール方法

インストール方法について説明する。

5.1 Unix 系 OS におけるインストール方法

juman-5.1.tar.gz を展開し, juman-5.1 ディレクトリに移動して, 次の手順を実行する。

1. './configure' を実行する。
 './configure' にオプションを与えることにより, インストール先やコンパイルオプションなどを変更することができる。詳細は, 同梱の INSTALL または './configure --help' の出力を参照のこと。
2. 'make' を実行する。
 JUMAN システムのコンパイルと辞書の構築が行われる。
3. (root で) 'make install' を実行する。
 JUMAN システムと辞書がインストールされる。インストールされる場所は以下のとおりである。\$PREFIX は, デフォルトでは /usr/local であるが, './configure --prefix' で設定できる。

<code>\$PREFIX/bin/juman</code>	実行ファイル
<code>\$PREFIX/share/juman/dic/</code>	辞書
<code>\$PREFIX/etc/jumanrc</code>	設定ファイル
<code>\$PREFIX/share/juman/doc/manual.pdf</code>	マニュアル
<code>\$PREFIX/libexec/juman/</code>	辞書作成プログラム
<code>\$PREFIX/lib/libjuman.*</code>	ライブラリ
<code>\$PREFIX/include/juman*.h</code>	ヘッダファイル

4. 品詞コストなどを個人で設定したい場合は、`$PREFIX/etc/jumanrc` を `$HOME/.jumanrc` としてコピーし、それを編集する。

5.2 Windows におけるインストール方法

`juman-5.1.exe` を実行し、表示される指示に従ってインストールを行う。以下のファイルがインストールされる。`$PREFIX` は、デフォルトでは `C:\Program Files\juman` であるが、インストール途中で設定できる。

<code>\$PREFIX/juman.exe</code>	実行ファイル
<code>\$PREFIX/dic/</code>	辞書
<code>\$PREFIX/dic/jumanrc</code>	設定ファイル
<code>\$PREFIX/manual.pdf</code>	マニュアル

現在の Windows 版は試験版であり未対応の機能がある。UNIX 版との違いは次のとおりである。

- 入出力は SJIS で行う (辞書などは UNIX 版と同様に EUC であるが、入出力の時点で漢字コードの変換をしている)。
- `jumanrc` での種々の設定はできない。辞書も一ヶ所 (1 フォルダ) しかもてず、ここに文法辞書と形態素辞書の双方をおく。辞書の場所の設定は `juman.ini` で行う。
- サーバ・クライアント機能は備えていない。

6 JUMAN の使用法

形態素解析の実行ファイルは `juman` という名前でインストールされている。オプション等の指定によって、この一つの実行ファイルをスタンドアロンモード (JUMAN3.4 以前のモード) とサーバ・クライアントモード (JUMAN3.5 以降で追加されたモード) で使い分けることができる。各モードの使用法を以下に説明する。

6.1 スタンドアロンモードでの使用

`juman` を実行する。標準入力から解析するテキストを一行一文として読み込み、解析結果を標準出力に出力する。このモードは、次節の方法でサーバ・クライアントモードを明示的に指定しない場合のデフォルトのモードである。

形態素解析に必要な文法・辞書の存在場所、なら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リソースファイルから読み込まれる。リソースファイルを探す順番は以下の順である。

1. 実行時に-r オプションで指定されたファイル .
2. ユーザのホームにある .jumanrc ファイル .
3. \$PREFIX/etc/jumanrc
4. 上記のいずれにもファイルが存在しない場合にはプログラムは強制終了される .

スタンドアロンモードでは以下のオプションが用意されている .

```
juman -[b|B|m|p|P] -[f|c|e|e2|E] [-i string] [-r rc_file]
```

- 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 .
 - B : -b オプションと同様に後方最長一致の解を表示するが , 同じ位置に複数の形態素がある場合はそれらをすべて表示する . ある位置からの 2 つ目以降の形態素は “@” をつけて表示する (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 .
 - P : -p と -B を合わせたもの . すなわち -p と同様に曖昧性の全組合せを示すが , 同じ位置の複数の形態素は -B のように “@” でまとめて表示する .
- 各形態素の表示方法
 - f : カラムを整えて表示 .
 - c : 完全な形態素情報をコードで表示 .
 - e : 完全な形態素情報を文字とコードで表示 .
 - e2 : -e の情報に加えて意味情報を表示 (default) .
 - E : -e の情報に加えて各形態素の文中での位置と意味情報を表示 .
- その他
 - i string : string で始まる入力行は解析せずにそのまま出力 .
 - r rc_file : rc_file をリソースファイルとして使用 .
 - h : ヘルプメッセージを出力 .

以下にいくつかのオプション指定による使用例を示す .

```
% juman
私は昨日学校を休んだ
私          (わたし)      私          普通名詞
は          (は)          は          副助詞
昨日        (きのう)      昨日        時相名詞
学校        (がっこう)    学校        普通名詞
を          (を)          を          格助詞
休んだ      (やすんだ)    休む        動詞          子音動詞マ行    タ形
EOS

% cat > test
子どもはリンゴがすきだ
```

```
% juman -B -e2 < test
子ども こども 子ども 名詞 6 普通名詞 1 * 0 * 0 "代表表記:子供"
は は は 助詞 9 副助詞 2 * 0 * 0 NIL
リンゴ りんご リンゴ 名詞 6 普通名詞 1 * 0 * 0 "代表表記:林檎"
が が が 助詞 9 格助詞 1 * 0 * 0 NIL
すきだ すきだ すきだ 形容詞 3 * 0 ナ形容詞 21 基本形 2 "代表表記:好きだ"
EOS
```

JUMAN5.0 以降、代表表記を設定し、意味情報として出力することとした (オプションとして `-e2` を指定)。詳細については E.12 節を参照。

6.2 サーバ・クライアントモードでの使用

JUMAN3.5 以降ではサーバ・クライアントモードでの使用が可能となった。このモードでは、あるマシンで形態素解析を行うサーバを起動しておき、他のマシン (同一マシンでも可) でテキストの入出力とサーバとの通信だけを行うクライアントを起動する。ネットワーク環境での複数ユーザによる利用などに適したモードである。

6.2.1 サーバの起動

オプション `-s` を指定して `juman` を実行することによりサーバとして起動される。

この場合のリソースファイルの探索順はスタンドアロンモードの場合と同様である。ただし、この時得られるリソース情報 (文法ファイルの位置、解析のコストなど) はクライアント側でリソースファイルが読み込まれない場合にのみ有効となる。

サーバに対するオプション指定は、サーバであることを示す `-s` オプションが追加されることを除いてスタンドアロンモードと同様である⁴。

6.2.2 クライアントの起動

オプション `-C hostname` を指定して `juman` を実行することによりクライアントとして起動される。hostname はサーバを起動したマシン名である。

もう一つの方法として、サーバを起動したマシン名を環境変数 `$JUMANSERVER` に設定しておく方法がある。この環境変数が設定されていればオプションの指定なしでもクライアントとして起動される⁵。

クライアントとして起動された場合のリソースファイルの探索順は次のようになる。

1. 実行時に `-r` オプションで指定されたファイル。
2. ユーザのホームにある `.jumanrc` ファイル。
3. 上記のいずれにもファイルが存在しない場合はサーバ側で指定されているリソース情報が使用される。

⁴サーバ・クライアント間の通信に用いられるポート番号のデフォルト値は 32000 である。この変更は `-N number` というオプションで行う。通常は変更の必要はない。

⁵サーバ・クライアント間の通信に用いられるポート番号がデフォルト値でない場合は `-C` オプションの場合も環境変数 `$JUMANSERVER` で指定する場合も、サーバのマシン名の後に (空白を入れずに) `'.'` をはさんでポート番号を指定する。

クライアントに対するオプション指定は、クライアントであることを示す-c オプションが追加されることを除いてスタンドアロンモードと同様である。

6.3 Perl からの利用

Perl モジュール “Juman” を用いることにより、Perl から JUMAN を利用することができる。

- インストール方法

JUMAN を展開したディレクトリにある perl ディレクトリに移動し、次の手順を実行する。

1. perl Makefile.PL
2. make
3. (root で) make install

- 使用例

プログラムの例を以下に示す。このプログラムを実行すると「この文を形態素解析してください。」という文の解析結果 (デフォルトのオプションは “-B -e2”) が表示される。

```
use Juman;
$juman = new Juman;
$result = $juman->analysis( "この文を形態素解析してください。" );
print $result->all();
```

詳細は ‘perldoc Juman’ の実行結果を参照のこと。

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，長尾真：「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。
- [4] 山地 治，黒橋 禎夫，長尾 真「連語登録による形態素解析システム JUMAN の精度向上」言語処理学会 第2回年次大会 1996.

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， \dots 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)
((普通名詞) ; 名詞は形態品詞であり，
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 接続規則辞書 (JUMAN.connect) の記述

A.4.1 接続規則の記述

```
接続規則      ::= ( ( 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 ) #接続コスト )  
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。

#接続コスト 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 接続規則の定義例

```
(( (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接続可能  
  (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, その接続のコストは 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 子音動詞力行)
    (意味情報 "walk(X)"))
  )
)
```

C リソースファイル (jumanrc) の記述例

(文法ファイル /usr/local/share/juman/dic)

(辞書ファイル /usr/local/share/juman/dic)

(品詞コスト

((*)	10)	
((特殊 *)	100)	
((動詞)	100)	
((形容詞)	100)	
((判定詞)	11)	
((助動詞)	10)	
((名詞 *)	100)	
((名詞 数詞)	40)	
((名詞 形式名詞)	70)	
((名詞 副詞的名詞)	70)	
((指示詞 *)	40)	
((指示詞 副詞形態指示詞)	60)	
((副詞 *)	100)	
((助詞 *)	10)	
((助詞 終助詞)	20)	
((接続詞)	100)	
((連体詞)	100)	
((感動詞)	100)	
((接頭辞 *)	50)	
((接尾辞 名詞性述語接尾辞)	14)	
((接尾辞 名詞性名詞接尾辞)	35)	
((接尾辞 名詞性名詞助数辞)	35)	
((接尾辞 名詞性特殊接尾辞)	35)	
((接尾辞 形容詞性述語接尾辞)	14)	
((接尾辞 形容詞性名詞接尾辞)	14)	
((接尾辞 動詞性接尾辞)	14)	
((未定義語 カタカナ)	5000)	
((未定義語 アルファベット)	100)	
((未定義語 その他)	5000)	

)

(接続コスト重み 4)

(形態素コスト重み 1)

(コスト幅 0)

D JUMAN サーバのプロトコルの概要

JUMAN サーバに接続すると、最初の 1 行に JUMAN やサーバのバージョンが出力される。現在の JUMAN では、入出力ともに、文字コードは日本語 EUC、改行コードは LF(ASCII 10) となっている。

JUMAN サーバと通信するためのコマンドを以下に示す。

RUN [options]

形態素解析を実行する。

RUN を、必要があれば後ろにオプションをつけて入力すると '200 OK' と出力される。オプションは JUMAN 起動時のものと同じである。

つづいて、日本語の文章を日本語 EUC で入力すると、改行を入力するごとにその行の解析結果が出力される。

入力は、Ctrl-k (ASCII 0b) のみの行を入力することで終了する。その結果、終了のしるしとして ASCII 0b と改行が出力されるが、これは普通の端末では空行 2 行が出力されたようになる。

解析中に異常が発生した場合は、'500' の後に、エラーメッセージが表示される。

RC

設定ファイルの読み込みを行う。

RC を入力した後、設定内容 (jumanrc ファイルの内容) を入力する。入力は、Ctrl-k (ASCII 0b) のみの行を入力することで終了する。

設定の読み込みが正常終了した場合は、'200 OK' と出力される。設定を誤った書式で入力した場合は、サーバプログラムは処理を中止し接続が切れる (より親切なエラーメッセージに変更の予定)。

HELP

簡単な説明を表示する。

QUIT

接続を切断する。

E JUMAN の更新履歴

E.1 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．
 - 接続コストを接続規則個別に定義できるようにした．
 - 形態素コストと接続コストに重みを付けることができるようにした．
 - 形態素間の接続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にした．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. 接続規則辞書中でマクロ定義を行えるようにした．
 7. 個人ユーザが記述するオプションな情報を .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．

E.2 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．

E.3 JUMAN 2.0 から JUMAN 3.0 への拡張点

(黒橋 禎夫 山地 治 大石 巧 坂口 昌子)

以下の機能拡張、および整備を行った．

1. 連語処理
2. 数詞の連結処理
3. 括弧、空白の透過処理
4. カタカナ語の扱い
5. 解析結果の表示オプションの追加
6. 標準辞書の整備
7. 標準文法の整備
8. 辞書システムの GDBM パージョンの追加

それぞれについて以下で詳しく説明する．

E.3.1 連語処理

形態素並びを連語として一括登録し、連語としての固有のコスト、固有の接続を指定できる機能を追加した。このことの利点は次の2点である。

- これまで品詞コスト、接続コストの調整では対処しきれなかった(対処すると副作用を生むような)解析誤りを、連語の登録によって解決することができる。連語登録にはほとんど副作用がない。
- 2語の接続可能性を指定するという、これまでの枠組では表現できなかった接続条件を、連語の接続条件として指定することができる。たとえば、ナ形容詞テ形に連語「は(副助詞) + ない(接尾辞)」が接続可能であるというような指定ができる。これを、ナ形容詞テ形と副助詞「は」、副助詞「は」と接尾辞「ない」がそれぞれ接続可能であるとしてしまうと大変な副作用がある。

連語の辞書記述は‘dic’という拡張子をもつファイルで行う。中間辞書、システム形態素辞書への変換はこれまでどおりである(makeint *.dic; makehd *.int とする場合、コンパイル作業はこれまでとまったく同じ)。辞書記述は次の形式で行う。

```
(連語
((副詞 ((読み より)(見出し語 より)))
(形容詞 ((読み よい)(見出し語 よい)(活用型 イ形容詞アウオ段)(活用形 *)))
) 0.6 )
```

はじめに形態品詞名に相当するものとして「連語」と指定し、次に連語を構成する各形態素の情報を通常の形態素辞書と同じ形式で並べる。連語内に活用する形態素がある場合は、その活用型と活用形を指定する必要がある(このとき見出し語と読みには基本形を与える)。連語の末尾が活用する形態素で、その活用形が何でもよいという場合(上の例で「よりよい」だけでなく「よりよく」、「よりよき」なども可能な場合)は“(活用型 *)”とする。

最後の数字は連語コスト ($0 < \text{連語コスト} \leq 1.0$) で、省略された場合は0.5となる。連語全体のコストは、連語を構成する各形態素のコストと、各形態素間の接続コストの総和に、この連語コストをかけた値として計算される。

一方、連語固有の接続規則を指定する場合は、次のように行う。

```
((形容詞 * ナ形容詞 ダ列タ系連用テ形))
((連語 * イ形容詞アウオ段 * はい))
(連語 * イ形容詞アウオ段 * もない)))
```

形態品詞名を「連語」、品詞細分類名を「*」とし、活用型、活用形は連語末尾の形態素の活用型、活用形を与える。連語固有の接続規則の指定がない場合には、左の接続については連語中の先頭の形態素の接続、右の接続については連語中の末尾の形態素の接続がそのまま使われる。

E.3.2 数詞の連結処理

入力文中に数詞(辞書に数詞として登録されている語)が連続して現れる場合、それらを連結して一語の数詞を出力する機能を追加した。これは標準文法依存の機能である。この機能を削除したい場合にはmakefile中のNEUMERIC_Pというフラグを削除すればよい。

E.3.3 括弧，空白の透過処理

括弧は接続する 2 語の間にほぼ任意に現れる，これを括弧とその前後の語との接続規則で規定することはほとんど不可能であった．そこで，括弧は任意の語の間に現れえるとして，その前の語と後の語の接続を調べるという透過処理の機能を設けた．たとえば“書いて」「いた”の場合，“書いて”と“いた”が接続可能であるかどうか調べられる．この時，加算されるスコアは“書いて”，“|”，“いた”の形態素スコア，および“書いて-いた”の接続スコアとなる．これも標準文法依存の機能であるので，この機能を削除したい場合には makefile 中の THROUGH_P というフラグを削除すればよい．

なお，通常は許されない接続が括弧が入ることで許されるようになるという現象もあるので，これまでどおり括弧を一形態素として扱う処理 (パス) も残し，そのような現象は括弧を通した接続規則として規定している (詳しくは接続規則を参照) ．

一方，丸括弧の場合は一つの括弧ではなく括弧のペア，“(…)” が任意に挿入される．この場合には“解(わか)る”のように 2 語の間ではなく 1 語の内部に挿入される用法もあるので，JUMAN の中で統一的に扱うことは極めて困難である．丸括弧を処理 (削除) するようなプリプロセッサを用意する方法で対処して頂きたい．(標準文法では一応 丸括弧も透過処理の対象としている)

テキスト中の 2 語の間に任意に現れる同じ問題として，空白の問題がある．2.0 版では入力文中の空白は自動的に削除され，形態素解析結果としては空白の存在は何も示されなかった．3.0 版では，空白についても接続の透過処理の枠組で扱うことに変更し，入力文中に空白があれば形態素解析結果としても空白が表示されるようにした．(この情報は形態素解析結果から元テキストを復元する際などに必要となる) ．

これに伴い，品詞「特殊」に「空白」という細分類を追加し (JUMAN.grammar)，全角空白は「特殊-空白」という品詞の形態素として辞書登録した (Special.dic) ．半角空白については，JUMAN が基本的に全角文字列に対する処理システムであり，半角文字列の辞書登録，解析をサポートしていなかったため，解析システム (プログラム) 内部で定義しておくという方法をとった．出力としては，全角空白はそのまま“ ”，半角空白は“\ ”と表示される．これらの空白処理の機能についても，makefile 中のフラグ THROUGH_P を削除することで無効にできる．

E.3.4 カタカナ語の扱い

これまでは入力文中のカタカナ連続を未定義語 (大きなペナルティが与えられる) として切り出していたが，これを通常のサ変名詞として扱うように変更した．また，標準辞書からカタカナのみからなる語を削除した．カタカナ表記の場合たえず新語がつくり出されるので，辞書で対処することは実質上不可能であると判断したためである．

E.3.5 解析結果の表示オプションの追加

以下の解析結果表示オプションとして，-B，-P，-E を追加した (従来からのオプションとともに 6 節で説明) ．

E.3.6 標準辞書の整備

語彙総数を増大し (異なり形態素数約 20 万)，また一つの形態素に対しても，次のように種々の表記の見出し語を用意した．

(名詞

(普通名詞

((読み ふりそで)

(見出し語 振り袖 (振りそで 1.6) (ふり袖 1.6) (ふりそで 1.6))))

しかし、平仮名を含む表記は解析誤りの原因ともなるので、名詞に関しては以下のような制限を設けた。

- 基本的な漢字 (小学校 1,2 年で習う漢字) を平仮名書きする表記は作成しない。
- 漢字の部分を平仮名書きする表記に対しては、その長さに応じて 1 文字の場合 3, 2 文字の場合 2.0, 3 文字以上の場合 1.6 という相対重みを与える。

3.0 標準辞書のコンパイル時間は SPARCstation20 上で約 2 時間半、作成される JUMANTREE.dbm.pag のサイズは約 130M バイトである。なお、もちろん JUMAN2.0 標準辞書を JUMAN3.0 システムで用いることも可能である。

E.3.7 標準文法の整備

標準文法の整備を行った。まず、接続規則の精密化を行った。特に、助詞間の接続については大幅な見直しを行った。

また、活用等に関しても、これまで扱っていなかった文語形、省略形 (口語的) などに対応した。

動詞	タ系連用チャ形	ちゃ	(書い <u>ちゃ</u> 困る)
動詞	音便条件形	りゃ	(泳ぎ <u>ゃ</u> 直る)
イ形容詞	文語連用形	ゅう	(美し <u>ゅう</u> ございます)
イ形容詞	文語基本形	し	(広 <u>し</u> といえども)
イ形容詞	文語命令形	かれ	(美しか <u>れ</u> と願う)
ナ形容詞	ダ列タ系連用ジャ形	じゃ	(きれい <u>じゃ</u> ない)
ナ形容詞	ダ列文語連体形	なる	(雄大 <u>なる</u> アフリカ)
ナ形容詞	ダ列文語条件形	なれば	(平和 <u>なれば</u> こそ)
ナ形容詞	デス列省略推量形	でしょ	(静か <u>でしょ</u>)
助動詞だろ <u>う</u> 型	デス列省略推量形	でしょ	(走らない <u>でしょ</u>)
動詞性接尾辞 <u>る</u> (活用型 母音動詞)			(食べて <u>る</u> , 食べて <u>た</u>)
動詞性接尾辞 <u>う</u> (活用型 子音動詞ワ行)			(書い <u>ちゃう</u> , 書い <u>ちゃった</u>)

判定詞に関しては特殊連体形「の」をたて、他の活用形もふくめて接続を正確なものに修正した。標準的に用意した.jumanrc(jumanrc.sample)では判定詞の品詞コストを 11, 助詞の品詞コストを 10 としているので、明確に判定詞である場合 (接続表で規定されている場合) は判定詞、それ以外は助詞と解釈される。たとえば「学生のように～」「学生ではない」「学生なわけがない」などは判定詞と解析される。

判定詞とナ形容詞関係の活用型の違いは次のようにまとめられる。

	基本連体形	特殊連体形	基本連用形
ナ形容詞	な (静かな)	—	に (静かに)
ナノ形容詞	な (大量な)	の (大量の)	に (大量に)
ナ形容詞特殊	な (同じな (ので))	* (同じ)	に (同じに)
判定詞	な (学生な (ので))	の (学生の (ようだ))	—

E.3.8 辞書システムの GDBM バージョンの追加

辞書システムのもとなっているハッシュ・ライブラリ NDBM は OS 依存で，移植性などに問題があったため，別に GDBM のバージョンを用意した .Makefile で # --- Use GDBM File ? 以下の 2 行をコメントアウトすれば NDBM，コメントアウトしなければ GDBM として動作する（配布時は NDBM）。

E.4 JUMAN 3.0 から JUMAN 3.1 への拡張点

1. 奈良先端科学技術大学院大学松本研究室の日本語形態素解析システム『茶筌』の辞書システム（パトリシア）を取り込んだ．これに伴いインストール方法（5 節参照），システム辞書名，プログラム名などに変更があるので，注意が必要．

JUMAN3.0		JUMAN3.1
JUMANTREE.table	↔	jumandic.tab
JUMANTREE.matrix	↔	jumandic.mat
JUMANTREE.dbm.dir		jumandic.dat
JUMANTREE.dbm.pag	↔	jumandic.pat
JUMANTREE.imis		
makehd	↔	makepat（正確には対応しない， 辞書のコンパイルは Makefile による）
.jumanrc の辞書ファイル指定		
ファイル名 (.../JUMANTREE)	↔	ディレクトリ名

2. 解析オプションとして，-i，-r を追加した（6 節参照）。
3. 固有名詞についてコスト調整を行い，ある程度正しく解析されるようにした．

E.5 JUMAN 3.1 から JUMAN 3.2 への拡張点

1. 従来，品詞定義として名詞の細分類に固有名詞，地名，人名があったが，標準辞書ではこれらを区別せずにすべて固有名詞としていた．これを地名，人名，その他（固有名詞）に分類した．
2. 助詞の細分類の中に名詞接続助詞と述語接続助詞があったが，この分類は不統一であったので，これらをまとめて接続助詞という細分類とした．また「と同時に」などの複合的な助詞を削除した．
3. 京都大学でのコーパス作成プロジェクトでの使用を通して，標準文法，標準辞書の整備を行った．品詞コストの調整も若干行ったので，個人用に .jumanrc を設定している場合は jumanrc.sample を参照のこと．

E.6 JUMAN 3.2 から JUMAN 3.3 への拡張点

1. 活用形に漢字が含まれる場合に (標準文法では「力変動詞来」と「動詞性接尾辞得る型」), その読みを活用型の定義に含めるようにした. これによって「来る」などの読みの部分に漢字が含まれることがなくなった.
2. 長い文を入力しても「Too many morphs」のエラーが出ないように改善した (形態素の配列の記憶領域の確保を動的なものに変更した).
3. 標準文法および標準辞書の整備.

E.7 JUMAN 3.3 から JUMAN 3.4 への拡張点

1. 固有名詞の分類として組織名をふやし, 固有名詞辞書を整備した. (SRI の亀山さん, NYU の関根さんに情報を頂きました. 感謝いたします)
2. 未定義語を品詞として JUMAN.grammar で定義し, 細分類として「カタカナ」「アルファベット」「その他」を設けた. 「カタカナ」はカタカナ列「アルファベット」はアルファベット列「その他」はそれ以外の同一文字種列 (ただし平仮名, 漢字は一文字) がプログラムによって切り出される. これらの接続, 重みなどは通常の品詞と同様に JUMAN.connect, .jumanrc などで定義する.
3. 標準文法および標準辞書の整備.

E.8 JUMAN 3.4 から JUMAN 3.5 への拡張点

1. サーバ・クライアント方式への拡張 (学術情報センターの影浦峽先生, 吉岡真治先生にご協力頂きました. 感謝いたします).
2. プログラム中のリソースファイルの扱いを以下のように整理した.
 - 関数 read_jumanrc_fn set_jumanrc_fileptr に変更
(リソースファイルのファイル名 (Jumanrc.File) ではなくファイルポインタ (Jumanrc.Fileptr) を設定)
 - 関数 read_jumanrcfile set_jumangram_dirname に変更
(文法ファイルのディレクトリ名を設定, 変数名は Jumanpath_rc から Jumangram_Dirname に変更)
 - 関数 read_jumanpathrc 削除
 - 関数 juman_init juman_init_rc に変更
(リソースファイル関係の初期化)
 - 関数 juman_init2 juman_init_etc に変更

E.9 JUMAN 3.5 から JUMAN 3.6 への拡張点

1. Windows 版の作成 .
2. Perl モジュール (Juman.pm) の作成 .
3. 連語処理の bug fix .

E.10 JUMAN 3.6 から JUMAN 3.61 への拡張点

1. iotool.c の修正 (\$HOME が設定されていなくて動作するように)
2. trans.c の修正 (Pentium II マシンで浮動小数の丸め誤差の影響について報告があったので対処)
3. 辞書・接続表などの若干の修正
(連語「しかない」「はしたものの」「においてよく」など)

E.11 JUMAN 3.61 から JUMAN 4.0 への拡張点

1. 辞書の変更: EDR 日本語単語辞書の利用をやめ、大規模テキスト中の統計量をもとに、約 6 万 8 千語の独自の辞書を構築した。これに関連して次の変更を行った。
 - (a) ナ形容詞語幹と名詞が重複するものについては、名詞見出し語を削除し、ナ形容詞語幹の接続を名詞に準ずるものとした。
 - (b) 一般的なカタカナ見出し語を与えたので「未定義語カタカナ」のコストを「未定義語その他」と同じく 5000 とした。これにより、例えば「テレビゲーム」は「テレビ」「ゲーム」となる。
 - (c) 形態素辞書の「意味情報」として”可能動詞”の記述を与えた。
2. juman.lib.c の bug fix

E.12 JUMAN 4.0 から JUMAN 5.0 への拡張点

概要

- 日本語の基本的語彙、約 3 万語 (固有名詞を除く) を選定した。
- 表記バリエーションの整備を行い、代用表記を出力することとした。
- その他の整備 (読みの音訓情報の付与「読ます」「読まされる」などの使役形への対応)

E.12.1 見出し語の整備

辞書の整備（表記バリエーション，意味情報付与など）を適切に行っていくために，また，不必要に大きな語彙による副作用（解析誤り）を防ぐために，標準的日本語で用いられる基本的語彙を選定した．ほとんど使われることのない古語などは排除し，複合語についても，構成性原理が成り立つものについては原則的に見出し語から排除した．判断基準は，京都コーパスでの頻度，新聞 2 千万文の自動解析での頻度，国語辞典の見出し語となっているかどうか，などから総合的に判断した．その結果，固有名詞を除く自立語（名詞，動詞，形容詞，副詞，連体詞，接続詞，感動詞）について，約 3 万語の辞書となった．

（ JUMAN4.0 からこの方向で整理を行ったが，今回，一応の整理を完成した．）

なお，京都コーパス 4.0 は JUMAN5.0 の語彙に対応している．

判断基準について

句，事典的な語は排除

例）育ての親，断腸の思い

元寇 倭寇 遣隋使 遣唐使 太郎冠者

連用形＋名詞も基本的には排除（相互情報量などで KNP で一語にする必要あり）

例）操り人形 打ち上げ花火 埋め立て地

2 文字のものは原則採用，3 文字以上のものは原則排除

例）印刷機，映写機，航空機 ...

観光地 原産地 ...

運動場 競技場 ...

組合員 銀行員 ...

観覧車 機関車 救急車 ...

無重力 無期限 ...

重金属 重工業 ...

しかし，以下のようなものは若干採用した．

- 例解小学国語辞典の見出しとなっているもの（の一部）

例）絵葉書

- 切り方が？なもの

例）工学部 医学部 全速力 水産物 海産物 林産物

- 音訓の原則（後述）では読みが分からなくなるもの

例）記念日

オレンジ色（青色 赤色 黄色 橙色などもあるので）

しかし，次のようなものは読みが分からなくなっても排除した．

例）基準日（にち）

金メダル 金メッキ（かね）

- 構成語が濁音となり、平仮名表記の場合に解析できなくなるもの。

例) 旗印(じるし) 溜まり醤油(じょうゆ) 掛け布団(ぶとん) 毒蜘蛛(ぐも)
計画倒れ(連用形でも濁音では解析できないので)

なお、構成語の濁音の平仮名表記は、一部、独立の見出し語とし、代表表記はもとの漢字表記とした。

例) じるし(印) じょうゆ(醤油) ぶとん(布団) げんか(喧嘩)

- カタカナの判断は難しい。例えば「フルタイム」「パートタイム」「フルセット」は排除、「フルコース」は採用した。
- その他、排除して問題となっているもの

例) 小(しょう)/悪魔
ロシア/正教(まさのり)
有田/焼(未定義語)
十(じゅう)/進法
トップ/10(いちぜろ)
トップテン(未定義語) ベストテン(未定義語)
一(いち)/手 一(いち)/局
万(まん)/国旗

- とりあえず採用したが?のもの

例) 添え

2文字の「～長」「～員」などは採用したが、関係解析などで「～」を独立させたい場合もあり、今後検討する。

例) (経理)部長(駒ヶ根)署員(民主)党员

動詞についてはまだ整理が不十分である。

例) 取り外す 取り付けるなどは採用している。しかし、可能形(取り外せる)はなく、2語に分割される。

E.12.2 自立語辞書の基本 format の整理

表記バリエーションとコストの扱い、それらの例外処理、代表表記、種々のコメントなどを一元的に管理できるように、自立語辞書の基本 format を定め、JUMAN 辞書はそこから自動生成することとした。

表記バリエーションは JUMAN3.0 以降自動生成していたが、例外処理などは、その結果に対してフィルタを用意しており、若干不透明であった。

また、表記バリエーション・代表表記の整理は今回新たに行ったものである。

この基本 format については、ドキュメントを整理中であり、別の機会に発表する予定である。また、基本 format のデータについては現在のところ公開していない。

E.12.3 表記バリエーションの整理

同じ語の表記バリエーションであることを扱うために、代表表記を設定し、これを意味情報（の一部）として出力することとした（-e2 オプション）。

解析例）

```
% cat sample.txt
子どもはリンゴがすきだ
かぜでおくれた

% juman -B -e2 < sample.txt
子ども こども 子ども 名詞 6 普通名詞 1 * 0 * 0 "代表表記:子供"
は は は 助詞 9 副助詞 2 * 0 * 0 NIL
リンゴ りんご リンゴ 名詞 6 普通名詞 1 * 0 * 0 "代表表記:林檎"
が が が 助詞 9 格助詞 1 * 0 * 0 NIL
すきだ すきだ すきだ 形容詞 3 * 0 ナ形容詞 21 基本形 2 "代表表記:好きだ"
EOS
かぜ かぜ かぜ 名詞 6 普通名詞 1 * 0 * 0 "漢字読み:訓 代表表記:風"
◎ かぜ かぜ かぜ 名詞 6 普通名詞 1 * 0 * 0 "代表表記:風邪"
で で で 助詞 9 格助詞 1 * 0 * 0 NIL
おくれた おくれた おくれる 動詞 2 * 0 母音動詞 1 タ形 8 "可能動詞 代表表記:送れる"
◎ おくれた おくれた おくれる 動詞 2 * 0 母音動詞 1 タ形 8 "代表表記:遅れる"
EOS
```

これによって、日本語処理の（初期段階における）最大の問題である表記揺れの問題を、形態素解析を行うだけである程度取り除くことが可能となる。

また、平仮名書き等による曖昧性も、適切な範囲で複数の可能性（代表表記）を挙げるようにしており、以後の構文解析・意味解析などへの適切な入力を提供することができる（上記の「かぜ」の例など）。

これらの整理は、不必要な大規模語彙について機械的に行っているのではなく、基本語彙について注意深く行っている。そのため、情報検索においても、形態素解析結果の代表表記のマッチングとすることで適合率・再現率を向上させることが期待できる。

この枠組みの中では、次のような現象を扱っている。

漢字と平仮名、送り仮名

漢字とするか平仮名とするか、また、送り仮名のバリエーション。その組み合わせ。（JUMAN3.0から扱っているが、代表表記を設け、再整理した）

```
例) 拳銃 けん銃 拳じゅう けんじゅう
    表す 表わす あらわす
    落とす 落す おとす
```

基本的なアルゴリズムは以下のとおり。

- 名詞、ナ形容詞以外はすべてのバリエーションを作り、コスト（ペナルティ）はない。

- 名詞，ナ形容詞は（完全平仮名表記以外で）小学 1～3 年生学習漢字を平仮名書きする表記は作らない．また，漢字を平仮名書きした表記にはコストを与える．
- 漢字 1 字 + 平仮名 1 字の表記は（平仮名部分のものの漢字が常用漢字であれば）ほとんど使われず，副作用のものとなので作らない

例）開き（開基） 合い（合意） 厚み（厚身）

漢字別表記

例）狩人／獵人 朝日／旭
色取る／彩る 哀れむ／憐れむ
綺麗だ／奇麗だ 気掛りだ／気懸かりだ

動詞でどこまでまとめるかは微妙な問題であり，一般の国語辞典などでは意味の一致の観点からはまとめすぎである．例えば「下りる／降りる」「下ろす／降ろす」はまとめないこととした．

カタカナ表記のバリエーション

ソフトウェアとソフトウエアのようなカタカナ表記のバリエーションに対しても代表表記を設けた．これは，カタカナ語のコーパス中の出現頻度をもとに，表記バリエーションの自動認識と複合語の自動分割を行い，その結果を人手でチェックして整理した．

日本語固有語のカタカナ表記

動物，植物，食べ物などで，日本語固有語であってもカタカナ表記されるものについて，代表表記のもとに整理した．

例）大根 だいこん ダイコン
餃子 ぎょうざ ギョウザ ギョーザ（最後はカタカナ表記バリエーション）

上記のカテゴリに関わらず，カタカナ表記があるものをまとめた

例）溝 ミゾ みぞ
眼鏡 メガネ めがね
奴 ヤツ やつ

代表表記は，原則として，新聞記事での高頻度表記としたが，この選択には強いこだわりはない（妥当性を主張するものではない）．重要なことは，同じ語の表記集合がまとめられ，これを通してテキストマッチングなどが適切に行われることであり，代表表記はこの集合の ID の役割を果たすものである．しかし，ID を数字などにすることは人間にとって管理しやすいものではないので高頻度の表記を採用した．ただし，現在の代表表記は ID として一意性を持つものではなく，これについては今後さらに整理する予定である．

なお，読みが複数ある漢字表記に対しては現時点では何も特別な扱いはしていない．

例）「旅客機／りょかくき」と「旅客機／りょかつき」は別見出し語
ただし代表表記はともに旅客機

E.12.4 読みの音訓の情報の付与

常用漢字 1 文字の見出し語に対して読みの音，訓の情報を与えた．

JUMAN では複数の読みを出す，KNP の最初で，独立の名詞，最初の名詞は訓読み，名詞に続く名詞は音読みを選択することで，大部分の読みは正しく扱える（この原則で対応できない場合は，一部，複合語として見出し語とした）．

例）運動場（じょう），救急車（しゃ），自由米（まい）

E.12.5 「読ます」の形の使役「読ませる」の形の使役受身への対応

「す」「さす」という動詞性接尾辞を作り対応した．

解析例）

読ま よま 読む 動詞 2 * 0 子音動詞マ行 9 未然形 3 "代表表記:読む"

す す す 接尾辞 14 動詞性接尾辞 7 子音動詞サ行 5 基本形 2 NIL

EOS

読ま よま 読む 動詞 2 * 0 子音動詞マ行 9 未然形 3 "代表表記:読む"

さ さ す 接尾辞 14 動詞性接尾辞 7 子音動詞サ行 5 未然形 3 NIL

れる れる れる 接尾辞 14 動詞性接尾辞 7 母音動詞 1 基本形 2 NIL

EOS

知ら しら 知る 動詞 2 * 0 子音動詞ラ行 10 未然形 3 "代表表記:知る"

す す す 接尾辞 14 動詞性接尾辞 7 子音動詞サ行 5 基本形 2 NIL

EOS

知ら しら 知る 動詞 2 * 0 子音動詞ラ行 10 未然形 3 "代表表記:知る"

さ さ す 接尾辞 14 動詞性接尾辞 7 子音動詞サ行 5 未然形 3 NIL

れる れる れる 接尾辞 14 動詞性接尾辞 7 母音動詞 1 基本形 2 NIL

EOS

以下，判断基準等の詳細

- 「読ます」は「読ま + す（接尾辞）」とする。「読ま + せる（接尾辞）」に近い「～ + さ + れる」とよく使うのはこれに相当する場合が多い．
- 「驚かす」「怒らす」「悩ます」「喜ばす」「悲します」など「人を～」のものも二語にする．
- 「知らす」「聞かす」も二語，ただし「知らせる」は使役度が極めて低く「伝える」に近いので一語としておく．
- 「励ます」は「励ませる」の意味と「元気づける」の意味があるが，後者が major なので，一語とする．
- 人が関係しない「乾かす」「（たばこ・大臣風を）吹かす」などは一語とする。「（肩を）いからす」も一語とするが，漢字表記の「怒らす」は「おこらす（二語）」への副作用があるのでやめる．
- 「（飛行機を）飛ばす」「（心を）踊らす」は一語とし「人を飛ばす・踊らす」は（本来二語としたいが）あきらめる．逆に「働かす」は普通人なので二語とし「頭を働かす」はあきらめる．

E.13 JUMAN 5.0 から JUMAN 5.1 への拡張点

1. 従来 KNP で付与していた語の意味情報を JUMAN で付与することに変更．(付属動詞候補，補文ト，～を～に構成語 など)
2. プログラムでの変数定義の修正 (MacOS X での問題解消)
3. 実行時のデフォルトのオプションを -B -e2 に変更
4. 辞書・接続表の若干の修正