语素分析系统 ChaSen 2.4.0 使用说明书

松本裕治　高岗一马　浅原正幸

2007 年 3 月

この翻訳版マニュアルは特定非営利活動法人言語資源協会の補助により作成されました。
此中文说明书由“特定非营利活动法人言语资源协会”协助制作。

Morphological Analysis System ChaSen 2.4.0 Users Manual
Yuji Matsumoto, Kazuma Takaoka and Masayuki Asahara

JUMAN	
version 0.6	17 February 1992
version 0.8	14 April 1992
version 1.0	25 February 1993
version 2.0	11 July 1994
ChaSen	
version 1.0	19 February 1997
version 1.5	7 July 1997
version 2.0	15 December 1999
version 2.2.0	06 December 2000
version 2.3.0	16 February 2003
version 2.4.0	30 March 2007
ChaSen for Windows	
version 1.0	29 March 1997
version 2.0	15 December 1999
version 2.4.0	30 March 2007
NAIST Technical Report	
1st edition(NAIST-IS-TR99008)	20 April 1999
2nd edition(NAIST-IS-TR99012)	15 December 1999

目录

前言

日文的语言解析处理与欧美的语言解析处理相比，首先存在以下两个问题。第一个问题是语素分析。计算机及文字处理系统的普及虽然明显的改善了日文的输入，但是在利用计算机来分析日文文本时，如何识别文本中的一个个的语素是我们需要首先面对的难点。为了进行语素识别，除了需要一个经久耐用的大规模词典之外，如何利用词典来进行语素识别也是不得不考虑的。另外一个问题是，虽然大家都很熟悉学校所教授的日文单词分类及语法用语，但是由于这些语言规则不适合用计算机进行处理，而且在研究者中也没有很高的评价，所以在日文中并没有一个被广泛认同的系统的语法体系。因此，在避免使用固定的语法体系的前提下来实现语素分析也是一个难点。

虽然很多研究小组经过长时间的技术研究已经开发出了相关的系统，但是对于日文解析处理最重要的语素分析系统，至今仍然还没有一个通用的工具。同样，计算机可读的日文词典也是如此。

因此，为了给广大的利用计算机来进行日文解析处理的研究者提供一个通用的语素分析器，我们开发出了这个 ChaSen 语素分析系统。在这套系统的开发过程中，由于我们考虑到了上述两个问题的存在，所以这个系统可以由使用者根据需求自行定义语法以及词语之间的连接关系。

这套系统是在大学中由少人数开发出的系统，因此一定很有很多不完善的地方。在以后的开发中，我们将会逐个改善不足点，希望大家可以谅解。

这套 ChaSen 系统的原型是 JUMAN(version2.0)。JUMAN 是京都大学长尾研究室和奈良先端科学技术大学院大学情报科学研究科合作开发的日文语素解析系统。在开发 JUMAN 的过程中，两所大学的教职工和学生给与了极大的支持和援助。此外，在词典方面，我们在 Wnn 假名汉字转化系统词典和由 ICOT 公开的日文词典的基础上，增加了独特的修改。在此，对共同开发 JUMAN2.0 的东京大学的黑桥祯夫和在佳能任职的妙木裕致以特别感谢。

感谢给与我们开发 JUMAN 的机会的长尾真教授。感谢在 JUMAN 的开发过程中给与我们各方面支持的筑波大学的宇津吕武仁教授。当时在奈良先端大就学的知念贤一给了我们关于 ChaSen 系统开发的很多建议。此外，当时在奈良先端大就学的今一修、今村友明、北内启，山下达雄，平野善隆及松田宽分别对 ChaSen1.0，ChaSen2.0 beta，ChaSen2.0 及 ChaSen2.2 做出了各种各样的贡献。在此，对他们两人及协助开发的松本研究室的成员致以深厚的谢意。由奈良先端大的鹿野清宏教授带领的[日文听写基本软件]的开发小组，对 ChaSen 系统中使用的 IPA 词性系统词典作了大幅的整理。特别感谢对开发给与极大支持的法政大学的伊藤克亘和 ASTEM 的山田笃。感谢在以口语处理为中心的词典构筑方面给与我们跟多建议的千叶大学的传康晴。当时在奈良先端大就学的高林哲夫，工藤拓在 autoconf、automake 化以及 RPM 包的建立方面给与我们了很多建议。此外，GOH Chooi-Ling、郑育昌、吕嘉在中文版词典的整理中给予了很大的帮助。在此，我们虽然不能列举出名字，但是我们同时也很感谢给与评论和提问的广大 ChaSen 的用户。

2007 年 3 月 30 日

关于本说明书有任何疑问，请与以下地址联系：
邮编：630-0192
地址：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 自然言語処理学講座
Tel: (0743)72-5240, Fax: (0743)72-5249
E-mail：chasen@is.naist.jp
URL：http://chasen-legacy.sourceforge.jp/

1 ChaSen 的使用

1.1 安装方法

1. 安装必要的工具。

   编译 ChaSen 时需要下面两个工具。

   - Darts[1]0.3 或者以后的版本
   - libiconv(如果系统的标准配置中没有自带的话)

2. 运行‘configure’

   ```
   % ./configure
   ```

   - 指定 Darts 的头文件时

     ```
     % ./configure --with-darts=/usr/local/include
     ```

   - 使用 libiconv 时

     ```
     % ./configure --with-libiconv=yes
     ```

   - 指定 libiconv 的目录时

     ```
     % ./configure --with-libiconv=/usr/local
     ```

   编译器(compiler)和编译选项(compile option)是自动设定的。configure的详细的使用方法请参照文件‘INSTALL’，或者参照帮助文件‘./configure --help’

3. 运行‘make’

   ```
   % make
   ```

   运行完后，会生成以下文件：ChaSen 自己的执行文件在chasen/chasen，库文件在lib/，词典生成程序在mkchadic/。使用 OS 自带的标准make，有时可能会编译失败，这时请改用 GNU make。

4. 运行‘make install’

   ```
   % make install
   ```

   ChaSen2.1 版以后，安装的路径有所变更。默认的安装路径如下所示：(PREFIX是可以用‘./configure --prefix’指定的，默认的路径是/usr/local)

PREFIX/bin/chasen	ChaSen 的执行文件
PREFIX/libexec/chasen/	词典生成程序
PREFIX/lib/libchasen.*	ChaSen 的库文件
PREFIX/include/chasen.h	头文件
PREFIX/share/chasen/doc/	说明文件

   下面的文件不被安装：

perl/ChaSen.pm	Perl 模块

chasenrc文件在安装 ChaSen 系统时并不被安装。在安装词典(≥ipadic-2.6.0 版)时，chasen-config从默认的chasenrc中得到路径，所以如果PREFIX/etc中没有charenrc文件时，系统将会自动拷贝文件到该目录。如果PREFIX/etc中有chasenrc文件时，由于不会被自动拷贝，故需要用户自行设定。

[1] http://cl.aist-nara.ac.jp/%7etaku-ku/software/darts/

1.2 使用方法

ChaSen 系统的执行文件，在‘make install’后会被自动安装在‘PREFIX/bin/chasen’。

- 语素的解析

  本 ChaSen 系统由如下的chasen命令开始运行

```
% chasen [option] [file]
```

  从标准输入或者从指定的文件中，以行为单位进行语素分析处理。

- 处理内容

  将计算成本(cost)的最小的结果(语素的各种分割方法和最小损耗的差在允许的范围内)，按照指定的选项格式输出。各种选项的意义请参照下一节。

- 使用举例

```
% cat temp
私は昨日学校へ行きました.
% chasen temp
私      ワタクシ    私      名詞-代名詞-一般
は      ハ          は      助詞-係助詞
昨日    キノウ      昨日    名詞-副詞可能
学校    ガッコウ    学校    名詞-一般
へ      ヘ          へ      助詞-格助詞-一般
行き    イキ        行く    動詞-自立          五段・カ行促音便    連用形
まし    マシ        ます    助動詞             特殊・マス          連用形
た      タ          た      助動詞             特殊・タ            基本形
.       .           .       記号-句点
EOS
```

1.3 使用时的选项

语素分析系统在运行时，有一些选项可供选择。下面总结出了这些选项，并加以解释。在用-r这样的后面带有参数的选项时，选项和参数中间的空格可有可无。

- 被分析的语句含有歧义时，输出的表示方法（当语句无歧义时，任何方法的输出结果都相同）

  -b 尾部最长一致的分析结果只输出一个（默认）
  -m 只对于含有歧义的部分用多个语素来表示
  -p 将歧义组合出的全部结果分别输出

- 各种语素在输出时的表示方法

  -f 按列对齐输出（默认）
  -e 所有的语素信息都用文字输出
  -c 所有的语素信息都用编码输出
  -v 为了使用 VisualMorhps 的详细的输出
  -F format 用format指定的形式将语素输出
  -Fh 显示-F选项的输出格式帮助

- 其它

  -j 将标点和空行作为断词符解析
  -o file 指定解析结果的输出文件
  -w width 指定计算成本(cost)
  -r rc_file 将rc_file作为chasenrc文件使用
  -R 读入默认的chasenrc文件(PREFIX/etc/chasenrc)
  -L lang 指定语言
  -lp 显示词性的名称及编号列表
  -lt 显示活用类型的名称及编号列表

-f 显示活用类型编号列表，活用形的名称和编号列表
-i 选择输入的文字编码(e:EUC-JP, s:Shift_JIS, w:UTF-8, u:UTF-8, a:ISO-8859-1)
-h 显示帮助信息
-V 显示 ChaSen 的版本号
-s 带制约条件的解析

-j 选项说明

通常情况下，ChaSen 以换行来判断一串文字列的结束点，因此，如果输入文字列的中间有空行的话，ChaSen 有可能不能正常进行解析。在这种情况下，如果使用了-j选项，ChaSen 会把标点(默认情况下包括“.。!?’’四个) 或者空行作为断词符来进行解析。而且，如果在 chasenrc 文件中的[断词符]项目中进行设定，用-j选项就可以将自己设定的文字作为断词符使用。

1.4 输出的格式

通过使用-F选项，或者在 chasenrc 文件的[输出格式]中指定输出的格式，可以改变解析结果的输出形式。

输出格式的文字串末尾如果有‘\n’，则输出有改行显示，并且在文末会自动添加‘EOS’。如果输出格式的末尾没有‘\n’，则所有的输出都会在一行显示，在文末才会有改行。

此外，在输出格式中指定‘-f’、‘-e’、‘-c’时，则分别按照-f、-e、-c的格式显示。

这里列出几个输出格式的例子：

- 和默认格式(-f)一样的输出

  “%m\t%y\t%M\t%U(%P-)\t%T␣\t%F␣\n’’或者“-f”

- 单词、读音、词性用 tab 分割表示

  “%m\t%y\t%P-\n”

- 只显示单词

  “%m\n”

- 显示单词间的分割(用空白区分单词)

  “%m␣”

- 汉字假名变换

  “%y”

- 注音表示。‘‘漢字(かな)’’的表示形式

  “%r␣()”

输出格式的变换文字如下表所示：

变换文字	功能
%m	单词(出现形)
%M	单词(基本形)
%y, %y1	读音的第一候补(出现形)
%Y, %Y1	读音的第一候补(基本形)
%y0	全部读音(出现形)
%Y0	全部读音(基本形)
%a	发音的第一候补(出现形)
%A	发音的第一候补(基本形)
%a0	全部发音(出现形)
%A0	全部发音(基本形)
%rABC	附带注音的表示(如：“A漢字Bかな C”)(注 1)
%i, %i1	附加信息的第一候补
%i0	附加信息的全体
%Ic	附加信息(空文字列表示，或者如果是“NIL”的话，用文字c表示)(注 1)
%Pc	用文字c来区分各级词性的文字列
%Pnc	用文字c来区分 1 n(n:1 9)级词性的文字列
%h	词性的编号
%H	词性文字列
%Hn	第n(n:1 9)级的词性(如果没有，则是最深一级的词性)
%b	0(仅是与旧版本的互换性)
%BB	词性的细分类(如果没有，则为该词性)
%Bc	词性的细分类(如果没有，则为文字c)(注 1)
%t	活用类型的编号
%Tc	活用类型(如果没有，则为文字c)(注 1)
%f	活用形的编号
%Fc	活用形(如果没有，则为文字c)(注 1)
%c	语素的计算成本
%S	解析全句
%pb	用“*”表示最优路径，用“␣”表示其他路径
%pi	路径的编号
%ps	路径的语素的开始位置
%pe	路径的语素的终止位置 +1
%pc	路径的计算成本
%ppiC	列举出用文字C区分的前方连接路径的编号
%ppcC	列举出用文字C区分的前方连接路径的计算成本
%?B/STR1/STR2/	用STR1表示词性的细分类，用STR2表示其他(注 2)
%?I/STR1/STR2/	附加信息不为“NIL”或者“”(空文字列)时，用STR1表示；否则用STR2表示(注 2)
%?T/STR1/STR2/	是活用时，用STR1表示；否则用STR2表示(注 2)
%?F/STR1/STR2/	与%?T/STR1/STR2/相同
%?U/STR1/STR2/	是未知词的话，用STR1\表示；否则用STR2表示(注 2)
%U/STR/	是未知词的话，用“未知词”表示；否则用STR2表示(与%?U/未知词/STR/相同)(注 2)
%%	符号%
.	指定字段的长度
-	指定字段的长度
1-9	指定字段的长度
\n	换行符
\t	制表符
\\	符号\
\’	符号’
\"	符号"

注 1 ipadic 中，当一个语素有多个读音的情况时(如[行く(いく/ゆく)])，我们用半角的括号和斜线来表示它的读音，如[ｲ/ユｸ]。一般的情况下，在读音的输出(输出格式%y中，只显示读音的第一候补[ｲｸ]，使用%y0的输出格式时，则可显示全部的读音[ｲ/ユｸ]。

注 1 当A、B、C、c为空白文字时，不显示任何内容。

注 2 在'/'处，可以使用任意的文字。此外，也可以使用括号"(){}[]<>"。示例如下：

- %?T#STR1#STR2#
- %?B(STR1)(STR2)
- %?U{STR1}/STR2/
- %U[STR]

1.5 附带约束条件的解析

[附带约束条件的解析]是指，在已经得知输入句子中一部分语素信息或者是分界位置的前提条件下进行的语素解析。

比如在句子[にわにはにわにわとりがいる。]中，可以在指定[はにわ]或者[にわとり]为一个语素的条件下进行解析。在这种情况下，与条件相异的"第 4 个文字[は]是一个单独语素"的可能性，或者将[にわとり]分解为[にわ]和[とり]等可能性将在解析是被排除。

输入格式： 附带约束条件的解析的输入格式与 ChaSen 的标准输出格式相同。但是读音、基本形的信息会被忽视。(\t 表示 tab)

```
にわ\|\t| ニワ\|\t| にわ\|\t|UNSPEC
に
はにわ\|\t| ハニワ\|\t| はにわ\|\t| 名詞-一般
にわとり\|\t| ニワトリ\|\t| にわとり\|\t|UNSPEC
がいる。
EOS
```

每一行都称为一个片段，一个片段由[语素段]、[句子断片]、[句末]、[注释]中的一种构成。

- 语素段

  表示这个片段是一个单一的语素(不能再继续分割)。

  在语素段的片段中，从第四列开始显示的是该语素的词性。词性的显示格式也和 ChaSen 的标注输出格式相同。

  如果没有显示词性，而显示的是[UNSPEC]，则表示在辞典中对这个片段做查找以后，并没有这个片段的解析结果。故作为未知词判断。

- 句子断片

  没有词性的片段用句子断片来表示。

  在这个片段内显示的文字列，在没有约束条件的情况下会照常被解析，只是解析生成的语素候补，不会出现跨越片段的情况。

- 句末

  由[EOS]，[BOS/EOS]，[句末]的符号开始的行，或者只有换行符号的行被成为句末。

- 注释

  词性信息的列显示[ANNO]的时候，表示这个片段为注释。

  注释的信息在输出是会显示出来，但是并不能被用在解析中。具体的显示格式请参照 chasenrc 文件。

解析例 下面对于具体举例说明解析的过程。

输入：

```
$ chasen -s
にわ\t ニワ\t にわ\tUNSPEC
に
はにわ\t ハニワ\t はにわ\t 名詞-一般
にわとり\t ニワトリ\t にわとり\tUNSPEC
がいる。
EOS
```

输出：

```
にわ\t\t\t 未知語
に\t ニ\t に\t 助詞-格助詞-一般
はにわ\t ハニワ\はにわ\名詞-一般
にわとり\t ニワトリ\t にわとり\t 名詞-一般
が\t ガ\t が\t 助詞-格助詞-一般
いる\t イル\t いる\t 動詞-自立\t 一段\t 基本形
。\t。\t。\t 記号-句点
EOS
```

带约束条件解析的注意点

- 在带有约束条件的解析过程中，如果在 chasenrc 文件中没有指定注释，即使在解析时指定“ANNO”，也不会输出任何内容。
- 在带有约束条件的解析过程中，空白词性功能和无视空白的功能被设为无效(用注释功能代替)。

2 chasenrc 文件

chasenrc 文件被用来定义语素解析系统中必要的各种各样的选项。这样的定义在通常情况下是记述在默认的设定文件(`PREFIX/etc/chasenrc`)中，但是用户在自己的 Home 目录的‘`.chasenrc`’文件中也可以进行定义。在启动系统是，用选项可以指定 chasenrc 文件。但是会按照下面的优先顺序来读入。

1. (Unix, Windows)启动时用`-r`选项指定的文件。
2. (Unix, Windows)中的环境参数`CHASENRC`指定的文件。
3. (Windows)的注册表路径`HKEY_CURRENT_USER\Software\NAIST\ChaSen`的 chasenrc 中设定的 chasenrc。
4. (Unix)用户的 Home 目录中的`.chasen2rc`。
5. (Unix)用户的 Home 目录中的`.chasenrc`。
6. (Unix)`PREFIX/etc/chasenrc`(默认不被安装)。

设定项目一览如下所示。其中，[`DADIC`]，[未知词词性]，[词性计算成本]必须要设定。

1. **语法文件的路径**

   指定语法文件(`grammar.cha, ctypes.cha, cforms.cha, connect.cha`) 所在的路径。

   ```
   (语法文件     /usr/local/lib/chasen/ipadic/dic)
   ```

   [语法文件]可以省略。在省略时，默认的路径和 chasenrc 文件所在的路径相同。ChaSen 语素解析系统的附带辞典(≥ipadic1.01 版)中的 chasenrc 文件默认省略[语法文件]。

2. **辞典文件**

   双数组的辞典文件(`chadic.{da,lex,dat}`) 是由文件名中去掉后缀名的前半部分指定的。辞典可以同时指定多个。此外，在指定路径时，如果使用了相对路径(不是用“/”开始的路径)，则辞典的默认路径和语法文件的路径相同。例如用下面的方法指定。

   ```
   (DADIC    chadic
             /home/rikyu/mydic/chadic)
   ```

   上面的指定方法，读入了下面两个辞典文件。

   (a) 和语法文件在相同路径中的`chadic.{da,lex,dat}`

   (b) `/home/rikyu/mydic/`中的`chadic.{da,lex,dat}`

在辞典检索的过程中，这两本辞典同时被使用[2]。

为了使用 Darts 的双数组辞典，需要指定[DADIC]

```
(DADIC    chadic)
```

上面的指定方法将会读入和[语法文件]在相同路径中的chadic.da，chadic.lex，chadic.dat文件。

可以使用的辞典的数量上限为 32 个。

3. **未知词的词性**

对于未知词，设定在发现未知词时，给该词赋予什么词性作为连接条件。指定多个词性时，各自对应其设定的连接条件。

```
(未知词词性 (词性 ｻ变接続))                    ;指定一个词性
(未知词词性 (词性 ｻ变接続) (名词 一般))      ;指定多个词性
```

4. **词性的计算成本**

在词素解析程序中，解析结果的优先顺序是按照计算成本的大小来决定的。当解析的过程中出现歧义时，计算成本总和最小的分支将被排在最优先的位置。在[词性的计算成本]中，定义了各个词性的计算成本的倍率和[未知词]的计算成本。各个计算成本的值为正整数。

```
(词性的计算成本
                ((*)     1)
                ((未知词)     500)
                ((名词)     2)
                ((名词 固有名词)     3)
)
```

对于同一词性的计算成本的定义，如果在计算的过程中进行了多次指定，则最后指定的计算成本优先。在上面的例子中，[名词]词性的计算成本一般为 2 倍，只有在[名词-固有名词]中的细分类的词性计算成本为 3 倍。此外，在开头指定‘(*)’的含义是，没有明确定义的语素的计算成本均默认为 1 倍(实际的计算成本)。对于未知词，计算成本均为 500 倍。

5. **连接计算成本和语素计算成本的相对权重的定义**

语素解析过程中的总计算成本，是由语素的计算成本和连接计算成本的总和决定的。如果想对这两种计算成本使用不同的权重时，可以分别进行指定。解析结果的总计算成本是这两种不同的计算成本与各自的权重相乘之后的总和。在省略时，默认权重为 1。

```
(连接计算成本权重   1)     ;默认值
(语素计算成本权重   1)     ;默认值
```

6. **计算成本的范围**

在语素解析的过程中，我们会想得到计算成本在一定的容许范围内的解析结果，而不是计算成本一直为最小值的解析结果。这个容许范围可以有用户指定。如果要输出计算成本在指定的容许范围内的所有解析结果，用户可以使用-m或者-p选项。

```
(计算成本范围   0)     ;默认值
```

计算成本的范围也可以用-w选项来指定。在这种情况下，用-w选项指定的范围优先。

7. **未定义的连接计算成本的指定**

用来指定在连接文件中没有被连接规则定义的语素之间的连接计算成本。如果未定义的连接计算成本没有被重新指定，或者被重新指定为 0 时，那么这意味着没有被连接规则定义的语素之间绝对不会有任何连接关系。这里的默认值为 0。

[2]虽然在一组辞典中不会多次登录相同的语素，但是在多个辞典的检索过程中，一个语素在多个辞典中同是出现的可能性存在。在这种情况下，系统会得到多次对于同一语素的信息。

```
(未定义的连接计算成本      500)
```

8. **输出格式**

   如果指定了输出格式，则可以更改解析结果的输出显示。

```
(输出格式      "%m\t%y\t%P-\n")
```

   输出格式可以用-F选项指定。在这种情况下，用-F选项指定的输出格式优先。详细请参照 1.4 节。

9. **BOS 文字列**

   用来指定解析结果的最开始显示的文字列。用“%S” 可以显示解析句子的全体。默认是空文字列(什么都不显示)。

```
(BOS 文字列      "解析文:  [%S]\n")
```

10. **EOS 文字列**

    用来指定解析结果的最后显示的文字列。用“%S” 可以显示解析句子的全体。默认是“EOS\n” 。

```
(EOS 文字列      "文末\n")
```

11. **空白词性**

    在 ChaSen 系统中，半角的空白文字(ASCII 编码 32)和 tab(ASCII 编码 9) 会被认为是空白文字，在实际的解析中会被忽略。通常情况下，解析结果中并不显示空白文字的信息，但是如果设定了[空白词性]，则空白文字的信息也将会再结果中列出。比如，按照下面的设定，空白文字将会输出[记号-空白]的词性。

```
(空白词性      (记号      空白))
```

    但是，如果在输出格式中使用了“%m”选项，而且指定了空白词性(什么词性都可以)后，输出的结果将和被解析的文字列完全相同。

12. **注释**

    以某个文字列开始，并以某个文字列结束的字串，可以被看作为注释，在解析的过程中无视。在最后的解析结果中，这个字串被看作是单一语素输出。

    在 chasenrc 文件中，定义了被看作注释部分字串的头文字列、末文字列以及字串的输出词性。其中，末文字列是可以省略的，在省略时，与头文字列相同的文字列将被作为注释使用。此外，如果省略了词性名或者格式字符串，那么被看作是注视的字串的信息将完全不会被输出。

```
(注釈  (("<" ">") "%m\n")
       (("「") (記号-一般))
       (("」") (記号-一般))
       (("\"" "\"") (名詞-引用文字列))
       (("［ " "］ "))
)
```

    例如，用上面的形式声明后，会以下面的说明的方式解析输出。

    - 像<img src="cha.gif">这样的，以三角括号“<”开始，以“>”结束的文字列，将会直接输出。
    - “「”或者“」”会输出「記号-一般」。
    - 像“"hello(again)"”这样的被双引号围起来的文字列会输出「名詞-引用文字列」。
    - 像“[ちゃせん]”这样的，以“[”开始，以“]”结束的文字列会在解析时被忽略，因此输出中并没有相对应的信息。

13. **连结词性**

在某种词性的语素连续出现时，将这些语素作为一个单独的语素输出时使用。

```
(連結品詞 ((複合名詞) (名詞) (接頭詞 名詞接続) (接頭詞 数接続))
          ((記号)))
```

比如，上面的例子可以用来连接下面的词性

(a) 连续出现的「名詞」「接頭詞-名詞接続」「接頭詞-数接続」用连结「複合名詞」表示。但是「複合名詞」的词性定义必须要在`grammar.cha`文件中明确写出才行。

(b) 将连续出现的「記号」连结起来，用单一的「記号」表示

14. **复合词的输出**

在语素词典文件(`.dic`)中定义的复合词，可以选择以下两种方式中的一种进行输出。第一种是输出整个复合词的语素信息(“复合词”)；第二种是输出构成复合词的各个单词的语素信息(“构成词”)。默认的输出方式是第一种(“复合词”)。

```
(复合词输出 "复合词")
```

此外，还可以用`-0c`, `-0s`选项来控制复合词的输出。

15. **分割文字**

使用`-j`选项可以指定解析时所使用的分割文字(参照1.3节)，分割文字既可以使用全角文字，也可以使用半角文字。

```
(分割文字    "。.、,!? .,!? ")
```

如果使用上面的例子中定义的分割文字，那么全角文字「。.、,!?」中的任何一个，或者半角文字“`.,!?`”中的任何一个都可以作为分割文字来使用。

16. **指定文字编码**

如果事先改变语素词典的文字编码并重新编译，那么就可以解析该文字编码的文件。这时在 chasenrc 中可以像下面的示例一样指定文字编码。

```
(ENCODE "w")
```

如果像上面一样进行定义，在输入 UTF-8 的文字编码的文件时，可以指定的文字编码有 e: EUC-JP, s:Shift_JIS, w:UTF-8, u:UTF-8, a:ISO-8859-1.

3 ChaSen 的库文件

由于 ChaSen 利用了`libchasen.a`，`libchasen.so`这样的库文件，所以我们可以将 ChaSen 的模块嵌入到其他的程序中。`chasen.h`文件作为头文件被安装。可以使用的库函数，变量如下所示。

```
#include <chasen.h>

int chasen_getopt_argv(char **argv, FILE *fp);

extern int Cha_optind;
```

- 将选项传递给 ChaSen。如果 ChaSen 还没有进行初期化，则在初期化进行之后开始设定选项。如果默认的选项可以使用时，可以调过这个函数而直接调用下一个解析函数。
- `argv`中指定了从命令行输入的以`NULL`结束的文字列数组。其中，`argv[0]`是程序的文件名。如果在指定选项时出现了错误，那么错误信息会输出到文件指针`fp`中。当`fp`是`NULL`的时候，什么也不输出。
- 制定选项出现错误时返回 1，没有错误时返回 0。

- 外部变量Cha_optind中保存了处理完毕的选项(包括argv[0])的个数。
- 下面举例进行说明。
  比如在chawan文件中，向 ChaSen 传递这样一个选项'-r /home/rikyu/chasenrc.proj -j'。在这个函数执行完毕后，Cha_optind中的值为 4。

```
char *option[] = {"chawan", "-r", "/home/rikyu/.chasenrc.proj", "-j", NULL};
chasen_getopt_argv(option, stderr);
```

```
#include <chasen.h>

int chasen_fparse(FILE *fp_in, *fp_out);

int chasen_sparse(char *str_in, FILE *fp_out);

char *chasen_fparse_tostr(FILE *fp_in);

char *chasen_sparse_tostr(char *str_in);
```

- 如果 ChaSen 还没有进行初期化，则在初期化完毕时候开始语素解析过程。根据输入和输出是文字列形式，还是文件形式的不同，有 4 个函数进行处理。
- chasen_fparse(), chasen_fparse_tostr()这两个函数从文件指针fp_in中读入文字列后进行解析。如果在chasen_getopt_argv()中指定了-j选项，那么句号等标点将会作为分割文字进行解析。
- chasen_sparse(), chasen_sparse_tostr()这两个函数进行文字列str_in的解析。
- chasen_fparse(), chasen_sparse()的解析结果会被输出到文件指针fp_out中。返回值为 0。
- chasen_fparse_tostr(), chasen_sparse_tostr() 将解析结果保存在 ChaSen 系统内部占用的内存中，并返回其地址的指针。这块内存领域在chasen_fparse_tostr(), chasen_sparse_tostr()被调用之前都会一直有效，其中的内容不会被释放。

4 在其它系统中的使用

4.1 在 Perl 编写的程序中调用

如果使用了perl/ChaSen.pm，就可以在 Perl 编写的程序中调用 ChaSen。具体的安装方法和使用方法请参照 perl/README。

参考文献

[1] 益岡隆志，田窪行則：「基礎日本語文法--改訂版--」， くろしお出版， 1992.

[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] 松本裕治，黒橋禎夫，宇津呂武仁，妙木裕，長尾真：「日本語形態素解析システム JUMAN 使用説明書 version 2.0」， NAIST Technical Report, NAIST-IS-TR94025， 1994.

[4] 山下達雄，松本裕治：「形態素解析視覚化システム ViJUMAN version 1.0 使用説明書」， NAIST Technical Report, NAIST-IS-TR96005， 1996.

[5] 山下達雄，松本裕治：「形態素解析結果の視覚化システム ViJUMAN とその学習機能」， 情報処理学会研究報告 96-NL-115, pp.29-34， September 1996.

[6] 平野善隆：「用言の活用を考慮した韓国語品詞体系の提案とそれを用いた韓国語形態素解析」， 奈良先端科学技術大学院大学修士論文, NAIST-IS-MT9551092， March 1997.

[7] 山下達雄：「規則と確率モデルの統合による形態素解析」， 奈良先端科学技術大学院大学修士論文, NAIST-IS-MT9551119， March 1997.

[8] 山下達雄，松本裕治：「コスト最小法と確率モデルの統合による形態素解析」， 情報処理学会研究報告 96-NL-119， May 1997.

[9] 北内啓，山下達雄，松本裕治：「日本語形態素解析システムへの可変長連接規則の実装」， 言語処理学会第三回年次大会論文集，pp.437-440， 1997.

[10]「研究開発用知的資源タグ付きテキストコーパス報告書」平成 9 年度， テキストサブワーキンググループ, 技術研究組合新情報処理開発機構， 1998.

[11] 松田寛：「品詞タグ付きコーパス作成支援環境の構築」， 奈良先端科学技術大学院大学修士論文, NAIST-IS-MT9851103， March 1999.

[12] 北内啓, 宇津呂武仁, 松本裕治：「誤り駆動型の素性選択による日本語形態素解析の確率モデル学習」， 情報処理学会論文誌 Vol. 40, No. 5, p.p.2325-2337, May 1999.

[13] 松田寛, 桐山和久, 山田悟史, 吉野圭一, 松本裕治：「部分形態素解析を用いたコーパスの品詞体系変換」， 情報処理学会研究報告 99-NL-134, p.p.23-30, Nov. 1999.

[14] Masayuki Asahara： Extended Statistical Model for Morphological Analysis, 奈良先端科学技術大学院大学修士論文, NAIST-IS-MT9851001， March 2000.

[15] 松田寛, 松本裕治：「品詞タグ付きコーパス作成支援 GUI ツール VisualMorphs」, 情報処理学会研究報告 2000-NL-137, p.98, June, 2000.

[16] 浅原正幸, 松本裕治：「統計的日本語形態素解析に対する拡張 HMM モデル」, 情報処理学会研究報告 2000-NL-137, p.p.39-46, June, 2000.

[17] Masayuki Asahara, Yuji Matsumoto： Extended Models and Tools for High-performance Part-of-Speech Tagger, Proceedings of COLING 2000, July, 2000.

[18] 浅原正幸, 松本裕治：「誤り駆動による統計的品詞タグづけモデルの拡張」, 情報処理学会研究報告 2000-NL-139, p.p.25-32, Sep. 2000.

[19] 松本裕治：「形態素解析システム『茶筌』」, 情報処理 Vol.41 No.11, p.p.1208-1214, Nov. 2000.

[20] 伝康晴，浅原正幸：「リレーショナル・データベースによる統合的言語資源管理環境」, 第 1 回「話し言葉の科学と工学」ワークショップ, Feb. 2001.

[21] 伝康晴，宇津呂武仁，山田篤，浅原正幸，松本裕治：「話し言葉研究に適した電子化辞書の設計」, 第 2 回「話し言葉の科学と工学」ワークショップ, pp. 39-46, Feb. 2002.

[22] 浅原正幸，松本裕治：「形態素解析とチャンキングの組み合わせによる日本語テキスト中の未知語出現箇所同定」, 情報処理学会研究報告, 自然言語処理研究会, SIGNL-154, pp.47-54, 2003

[23] 中川哲治，工藤拓，松本裕治：「Support Vector Machine を用いた形態素解析と修正学習法の提案」, 情報処理学会論文誌, Vol.44, No.5, pp.1354-1367, May 2003.

[24] Taku Kudo, Kaoru Yamamoto, Yuji Matsumoto： ‘‘Appliying Conditional Random Fields to Japanese Morphological Analysis”, EMNLP-2004, 2004.

[25] 松本裕治，高岡一馬，浅原正幸，工藤拓：「茶筌と南瓜による日本語解析--構文情報を用いた文の役割分類」 人工知能学会誌, Vol.19, No.3, pp.334-339, 2004.

[26] Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto： ”Chinese Word Segmentation by Classification of Characters”, International Journal of Computational Linguistics and Chinese Language Processing , Vol.10, No.3, pp.381-396, September, 2005.

[27] Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto： ”Training Multi-Classifiers for Chinese Unknown Word Detection”, Journal of Chinese Language and Computing, Vol.15, No.1, pp.1-12, 2005.

[28] ゴーチユイリン，鄭育昌，浅原正幸，松本裕治：「中国版茶筌の開発」, 言語処理学会第 11 回年次大会発表論文集, pp.245-248, 2005.

[29] 浅原正幸，高橋由梨加，松本裕治：「異表記同語情報を付与した辞書の整備」, 言語処理学会第 11 回年次大会発表論文集, pp.604-607, 2005.

[30] 工藤拓：「形態素周辺確率を用いた分かち書きの一般化とその応用」, 言語処理学会第 11 回年次大会発表論文集, 2005.

[31] Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto： "Machine Learning-based Methods to Chinese Unknown Word Detection and POS Tag Guessing", Journal of Chinese Language and Computing, Vol.16, No.4, pp.185-206, 2006.

[32] 東藍，浅原正幸，松本裕治：「条件付確率場による日本語未知語処理」 情報処理学会研究報告, 自然言語処理研究会, SIGNL-173, pp.67-74, 2006.

[33] 東藍，工藤拓，浅原正幸，松本裕治：「日本語未知語処理のための大規模未解析データの利用法」情報処理学会研究報告, 自然言語処理研究会, SIGNL-179, 2007.

附录

A 版权和使用条件

ChaSen 系统，是为了普及并支持自然语言处理研究而开发的免费软件系统。ChaSen 的版权由奈良先端科学技术大学院大学情报科学研究科自然言语处理讲座(松本研究室)保有。对于本系统的使用、更改和再发布，并没有特别的限制，但是在再发布本软件时，必须遵守以下条件。

- 再发布软件时，必须包含本节的有关于版权的说明和本使用说明书首页反面的有关于版权的声明。

奈良先端科学技术大学院大学做为本软件系统的版权拥有者，对于使用本软件或者基于本软件的再发布软件时所造成的一切损失，不承担任何责任。

上述的有关于版权的说明仅仅是 ChaSen 软件系统本身的声明，对于 ipadic 等其它的辞典，请参照它们自身的版权声明。

B 更新履历

B.1 ChaSen2.3.3 到 ChaSen2.4.0 的变更点

- 带有制约条件的解析的实现
- Windows 版软件包的再组装
- 用 chasenrc 指定文字编码的实现
- 追加用'u'指定 UTF-8 编码的功能

B.2 ChaSen2.3.2 到 ChaSen2.3.3 的变更点

- 辞典中没有读音和发音的信息时，%y、%a 等用空白表示
- 用注册表来指定 chasenrc 和语法文件的路径(MinGW 版限定)

B.3 ChaSen2.3.1 到 ChaSen2.3.2 的变更点

- 辞典检索的高速化
- 支持活用形指定辞典

B.4 ChaSen2.3.0 到 ChaSen2.3.1 的变更点

- PATDIC，SUFDIC 的废除
- `-i`选项(选择文字编码)的增加
- 支持 UTF-8 编码

B.5 ChaSen2.2 到 ChaSen2.3 的变更点

- 使用双数组 Darts 的辞典的构筑
- 服务器、客户端模式的废除
- 命令行解释器的废除
- 使用`cforms.cha`编辑定义为基本形的文字列功能的实现

```
(基本形　基本形-一般)
```

B.6 ChaSen2.0 到 ChaSen2.2 的扩展点

- 辞典和系统的分离

  为了完备其它语言的辞典，将辞典与软件系统分离。chasenrc文件由辞典部分保存，在安装系统是并不被安装。在安装辞典时，需要安装PREFIX/etc中的chasenrc文件。

- autoconf, automake, libtool 化

  在./configure 中，实现了可以设定自动读入环境变量的功能。同时，追加了在编译各个辞典时输出必要信息的程序chasen-config。

B.7 JUMAN2.0 到 ChaSen2.0 的扩展点

在 ChaSen2.0 中增加了词性系统和连接规则的机能。这个机能扩展版称为 v-gram 版，以前的版本称为 bi-gram 版。由于 v-gram 版和 bi-gram 版的语法文件的格式不同，因此它们所使用的辞典不能互换。但是，如果执行mkchadic/convdic命令，可以将 bi-gram 版的辞典转换为 v-gram 版的辞典。

convdic命令在 bi-gram 版的辞典目录中，用 v-gram 版的辞典目录作为参数执行。比如用下面的命令，在 bi-gram 版辞典的目录dic中会创建一个dic2目录，在dic2目录中存放 v-gram 版的辞典。但是，在执行convdic后，需要将附属于 ChaSen 的dic/Makefile 拷贝到 v-gram 版辞典的目录中(在下例中dic2)。此外，还需要准备chasenrc文件。

```
% cd dic
% ../mkchadic/convdic ../dic2
% cp Makefile ../dic2
```

ChaSen2.0 在默认情况下会编译 v-gram 版。如果使用‘make bigram’命令，则可以编译 bi-gram 版，以备使用。

v-gram 版和 bi-gram 版相比，有下面的扩张点和变更点。

1. 词性从两层定义扩张到多层定义。
2. 连接规则从 bi-gram 的固定固定长度扩张到 variable-gram(可变长度)。换句话说，不但可以记述两个连接的单词(或者词性)的连接计算成本，而且还可以记述 3 个以上的任意长度的单词(词性)列中的单词(词性)的连接计算成本。
3. 利用*.dic可以使用[发音]属性。输出格式中可以表示%a和%A。此外，使用 cforms.cha 可以定义发音的词尾。
4. 利用*.dic可以使用[base]属性。在显示所查找词语的基本形时，如果该词语有活用形，则显示其基本形；如果该词语没有活用形而有 base，则显示其 base。可以使用英文等辞典。
5. 扩张了chasenrc文件的[连结词性]机能，可以分别连结多种类词性。
6. 对于空行也显示“EOS” (正确情况下应该是BOS文字列和EOS文字列)。即“EOS”’的个数和输入文的行数相同。
7. 在解析结果的默认输出格式(-f)中，所查找词语后的分割用 Tab 替代空格。
8. 没有在辞典中登录的词语用[未知词]代替以前的[未定义词]来表示。
9. 在语素辞典*.dic中，单词的计算成本未被指定时，在 bi-gram 版中，使用默认的计算成本 10；在 v-gramn 版的*.dic中，使用[默认词性计算成本]指定的计算成本值(未被指定时使用 65535)。
10. 在 bi-gram 版中，语素计算成本和连接计算成本在内部使用 10 倍；v-gram 版中使用默认值。此外，bi-gram 版中的语素计算成本的范围是 0 655.3 (ChaSen1.51 以前是 0 25.5)；v-gram 版中范围是 0 65535。
11. 连接计算成本 0 指[以 1 的概率连接]，-1 指[不连接]。此外，连接计算成本的范围变为-1 32767。
12. 带有分割句子片段的功能，可以使用长度为 0 的词性。在词性定义文件中，在词性名之后加上‘’即可使用分割句子片段的功能。

B.8 ChaSen1.5 到 ChaSen2.0 的扩展点

再次仅列出 v-gram 版和 bi-gram 版的共同扩张点。

1. 省略了 chasenrc 的[语法文件]。如果[PATDIC]和[SUFDIC]不是以‘/’开始的，则认为它们与[语法文件]所在目录的相对路径相同。
2. 在辞典检索中使用了 SUFARY，实现了对于半角文字的检索功能。
3. 利用 SUFARY 还可以对英文进行解析。
4. 如果不是用-D 选项而使用-R 选项，则会读入由 Makefile 指定的 chasenrc 文件。
   (/usr/local/share/chasen/dic/chasenrc 等)
5. 实现了设定文头及文末的输出文字列的功能。
6. 实现了指定多个未知词及其计算成本的功能。
7. 实现了在 chasenrc 文件中指定[空白词性], 从而可以在解析结果中输出空白信息的功能。
8. 实现了在 chasenrc 文件中指定[注释]，从而可以在解析中像忽视空白一样忽视 SGML 等标注信息。
9. 实现了利用-lp、-lt、-lf 选项来显示词性及活用列表的功能。
10. 实现了利用-o 选项来指定输出文件的功能。
11. 实现了输出格式中可以使用"%?T/STR1/STR2/"的个功能。活用时输出STR1，非活用时输出STR2。此外，还可以在输出格式中使用%?I、%?B、%?F、%?U。
12. 实现在输出格式中使用"%rABC"来显示注音的功能。
13. 实现了在 chasenrc 文件中指定[BOS 文字列]和[EOS 文字列]，从而可以设定文头和文末的输出文字列的功能。
14. 实现了在 BOS 文字列、EOS 文字列和输出格式中用"%S" 来显示整个解析句子的功能。
15. 辞典文件中的语素计算成本的范围由 0 25.5 变为，bi-gram 版的 0 6553.5 和 v-gram 版的 0 65535。
16. 连接文件中的连接计算成本的范围由 0 255 变为 0 3267。

B.9 ChaSen1.0 到 ChaSen1.5 的扩展点

1. 进行库文件化，实现了从其他程序中可以简单的调用 ChaSen 模块。
2. 进行服务器与客户端化，实现了从其他的客户端机器上可以进行解析的功能。此外还制作了 Emacs Lisp 的客户端用户界面。
3. 实现了利用-w选项来指定计算成本范围的功能。
4. 实现了在chasenrc文件中指定[分割文字]，从而可以设定 jfgets()的分割文字的功能。而且，还可以指定半角文字。此外，默认的分割文字变更为“. 。！？ ”。
5. 由于确保了缓存动态保存数据的功能，在解析长的文字列时，不会出现“Too many morphs”的警告。
6. 在 ViCha 中新增了-v 输出选项。
7. 实现了在同时指定-d和-b选项时，只输出用-d格式表示的最优路径。

B.10 JUMAN2.0 到 ChaSen1.0 的扩展点

1. 辞典检索的方法由通常的利用 NDBM 来模拟 TRIE 构造的实现方法，变更为独自开发的利用 patricia 树的实现方法。解析时所需要的辞典大小变为了原来的四分之一。而且，编译辞典所需要的时间变为了 3 40 分之一。
2. 重新改造了解析系统，实现了高速化。解析速度变为原来的 8 11 倍(与 JUMAN2.0 相比)。
3. 重新编写代码，是系统可以安装到多种平台上。此外，还实现了利用 GNU 的 C 编译器(gcc)之外的，操作系统自带的 C 编译器进行编译的功能。
4. 实现了日语 EUC 之外的，JIS(ISO-2022-JP)的文字列的解析功能。
5. 由于导入了未定义连接计算成本，减少了未定义词语的输出。
6. 利用连接词性的定义，在输出最优路径时，实现了被定义的词性的单词被连接为一个单词的输出功能。
7. 利用定义活用词尾的读音的功能，可以实现用片假名来表示[来る]、[得る]等单词的读音的功能。
8. 追加了分割选项(`-j`)，实现了用句号断行而不是用改行编码断行的功能。
9. 实现了利用`-r`选项或者环境变量`CHASENRC` 来指定`chasenrc`文件的功能。
10. 实现了利用`-F`选项或者`chasenrc`文件中的[输出格式]来指定解析结果中的输出格式的功能。
11. 重新对语法进行定义，将词性分类[特殊]中的[括弧]划分为[括弧开]和[括弧闭]。此外，在[特殊]中有增加定义[空白]，具体用来表示全角空白。
12. 在助动词的活用类型中增加了[助動詞べきだ型]。助动词[べきだ]的活用由一直以来的[ナ形容詞]型变更为[助動詞べきだ型]。
13. 重新审视辞典中登录的词性，对它们做了一定的增减修正。

C JUMAN3.0 与 ChaSen 的关系

JUMAN 2.0 在 1994 年 7 月公开发布以后，京都大学长尾研究室和奈良先端大松本研究室分别朝着不同的方向尝试着进行了扩展。在京都大学，为了对通常的 bi-gram 模型不能处理的连接关系进行描述，通过增加连语处理和括弧透过处理等功能和大幅修正语法文件及语素辞典的方式对原系统做了扩展。在奈良先端大，由于考虑到今后日语标注语料库的规模不断增大，采用了增加从语料库中自动学习 bi-gram 以上的连接规则的功能(包含单词级和词性级的设定) 和使用不依存于 UNIX 的哈西数据库 NDBM 的辞典构筑思想的方式，对原系统进行了扩展。后者的扩展主要是为了响应在 UNIX 以外的操作系统中运行的用户要求，和改善辞典的编译时间及检索速度为目标进行的尝试。由于在这两种扩展系统中，对 bi-gram 以上的连接规则的处理有着很大的差异。为了融合两者的意见，京都大学在 1996 年 6 月提前发布了 JUMAN3.0beta 的扩展版。

下面列出了奈良先端大对于原系统的预定扩展项目。ChaSen1.0 在 1997 年 2 月公开，此后，经过 ChaSen1.5、1.51、2.0、2.2、2.3，最后到 ChaSen2.4，基本上全部实现了预定的功能。

1. (ChaSen1.0) 辞典系统的独自开发(去除 NDBM，采用 patricia 树)
2. (ChaSen1.0) 解析系统的重视与高速化
3. (ChaSen1.0) 未定义连接计算成本、连接词性、解析结果的输出格式的导入
4. (ChaSen1.0) JIS 文字列的解析
5. (ChaSen1.0) 活用词尾的读音的定义
6. (WinCha1.0) 在 Windows 系统中的应用
7. (ChaSen1.5) 库文件化
8. (ChaSen1.5) 服务器化
9. (ChaSen2.0) 词性定义的多层化

10. (ChaSen2.0) 连接规则的可变长化
11. (ChaSen2.0) 在辞典中登录包含半角文字的单词(利用 SUFARY 的辞典)
12. (ChaSen2.0) 输出格式的扩充
13. (ChaSen2.0) 从解析完毕的数据中对可变常接续计算成本的学习
14. (ChaSen2.4) 带有制约条件的解析

D 语素解析系统的发展

工藤拓氏公开了名为 Mecal 的语素解析系统。[3]。相对与 ChaSen 所采用的 HMM 模型(Hidden Markov Model)，MeCab 采用了 CRF 模型(Conditional Random Field)。工藤氏的论文[24] 证明了新的模型可以提高解析的准确度。可以输出[软分割] [?][4] 是 MeCab 的另一个特点。在目前的 ChaSen 的设计中，不能像 MeCab 那样自由定义属性(特征量)，因此不能对应这样的新模型。

对于解析用的辞典，近年也有了很多改良。新的 JUMAN 辞典[5] ，不但可以选择日语的基本词汇，还完备了很多动态表示信息。即将发布的由千叶大学的伝氏小组制作的 UniDic [21]，是不但可以被自然语言处理界的研究者使用，而且人文系的研究者和声音处理学的研究者也可以使用的易用的辞典。在奈良先端大，以后将会公布由经过分类的 IPADIC 辞典项目，和添加了动态表示信息及复合词信息的日语辞典。这个新辞典的名字将会变更，而且将会废除在 IPADIC 中还未解决的 ICOT 项目。此外，在奈良先端大，在处理好版权关系之后，将会公布标注完毕的基于 Penn Chinese Treebank 词性体系的中文语素解析系统辞典。在公布中文语素解析辞典时，会于 MeCab 的作者工藤氏商谈，不单是公布 ChaSen 用的模型，同时也公布 MeCab 用的模型。

JUMAN、ChaSen、MeCab 还没有解决的共同问题是未知词(辞典中没有登录的词)的处理问题。现在，奈良先端大正在开发解决未知词问题的机器学习的模型[32, 33]。无论如何扩展，我们期望能在将来公开与现在的 ChaSen 系统完全不同的，带有解析未知词模型的语素解析系统。

[3]http://mecab.sourceforge.net/
[4]http://mecab.sourceforge.net/soft.html
[5]http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html