载《语文研究》2006 年，第 3 期（总第 100 期）

机器翻译词典中语言信息的形式表示方法[1]

Formal Representation of Linguistic Information in Machine Translation Dictionaries

冯志伟
（教育部语言文字应用研究所）

摘要：机器翻译词典必须是计算机可读的，这样的词典应当使用形式化的方法表示出来。文本根据 MMT 模型中的复杂特征理论，分别研究了在机器翻译的分析词典、转换词典和生成词典中语言信息的形式化表示方法，如“宏表示法”“特征组合”“条件过程”“赋值过程”等。这些形式化表示法都是建立在复杂特征的基础之上的。

关键词：机器翻译，词典，形式表示，MMT 模型，复杂特征。

我曾经设计过若干个机器翻译系统，例如，汉-法/英/日/俄/德多语言机器翻译系统 FAJRA，英-汉机器翻译系统 ECMT，法-汉机器翻译系统 FCAT，德-汉机器翻译系统 GCAT，日-汉机器翻译系统 JCMT ，英-日机器翻译系统 E-to-J 等。在研制这些机器翻译系统的过程中，我提出了“多叉多标记树形图模型”（Multi-branch and Multi-label Tree Model，简称 MMT 模型），MMT 模型采用“多标记”（也就是“复杂特征”）来表示语言信息。在本文中，我们将根据 MMT 模型中的复杂特征理论，采用复杂特征这种适合于计算机处理的形式来描述语言信息，把机器翻译词典中的语言信息形式化地表示出来。

词典是机器翻译系统中规模最大的组成部分，也是机器翻译系统最重要的语言知识资源。机器翻译词典开发的工作量很大，需要大量的人力和时间。机器翻译系统的性能，在很大的程度上依赖于词典的质量。使用低质量词典的机器翻译有如做无米之炊，这样的翻译系统的性能肯定是十分低劣的。

有一些机器翻译系统的词典向用户开放，用户可以根据自己的需要增删词典的内容，所以，一般用户也有必要了解机器翻译词典的知识。

机器翻译词典有三种：分析词典、转换词典、生成词典。有的机器翻译系统同时具有这三种词典，分别用来做分析、转换和生成。有的机器翻译系统把分析词典和转换词典合并，另有一部生成词典用来做形态生成，这些机器翻译系统只有两部词典。有的机器翻译系统把分析词典和转换词典合并成一部词典，形态生成通过机器翻译系统的规则来实现，这样，这些机器翻译就只有一部词典了。为了叙述上的方便，我们这里仍然分别来讨论分析词典、转换词典和生成词典。

1　分析词典中语言信息的形式表示方法

机器翻译的分析词典是在源语言自动分析中使用的词典。分析词典中词汇单元所包含的特征应该是多方面的。尽管由于机器翻译系统的目的不尽相同，不同的机器翻译系统对于词汇的描述各具特色，但是，各个系统几乎都要具体地描述词的词类特征、词的次类特征、词的性、数、人称、时态、体、语气、语态等等。如果要作较为深入的自动剖析，还需要描述单词的语义特征；在许多有屈折变化的语言中，除了描述单词形态的规则的屈折变化之外，还需要描述单词形态的不规则屈折变化。一般来说，这些特征都可以使用复杂特征（complex features）来描述。

分析词典的词汇中所包含的纯句法信息主要有三种类型：

(1)词类特征：例如，某词为动词，某词为名词等；

(2)词与词之间的结合特征：例如，某词的主语是什么，某词的补语是什么等；

(3)与句法有关的词的其他特征：例如，名词的性、数等。

[1] 本文得到国家社会科学基金资助，项目号 03BYY019

这三种类型的纯句法信息，在基于特征的句法分析中是用词的句法范畴来表示的。例如，德语 Mädchen（姑娘）的句法信息可表示为：

lexeme Mädchen:

<cat> = N

<gender> = neut

其中，<gender>表示“性”，其值 neut 表示“中性”（neutral）。因此，这个词汇条目表示德语的 Mädchen 是一个中性名词。

英语 love（爱，喜欢）的句法信息可表示为：

lexeme love:

<cat> = V

<arg0 cat> = NP

<arg0 case> = nom

<arg1 cat> = NP

<arg1 case> = acc

其中，<arg0 case>表示“论元 0 的格”，其值 nom 表示“主格”（nominative），<arg1 case>表示“论元 1 的格”，其值 acc 表示“宾格”（accusative）。因此，这个词汇条目表示英语 love 是一个动词。这个动词具有一个主格主语 NP 和一个宾格宾语 NP。我们用 arg0（论元 0）表示主语，用 arg1（论元 1）表示直接宾语。

英语 give（给）的句法信息可表示为：

lexeme give:

<cat> = v

<arg0 cat> = NP

<arg0 case> = nom

<arg1 cat> = NP

<arg1 case> = acc

<arg2 cat> = PP

<arg2 pform> = to

其中，PP 表示介词词组，<arg2 pform>表示论元 2 这个介词词组的介词形式（pform）是 to。

英语 bet（打赌）这个动词使得我们还得使用 arg3 来表示论元 3。例如，在句子

He bets me ten dollars on John's coming.（他认为约翰会来，与我打赌十元。）

其中，he 是 arg0，ten dollars 是 arg1，me 是 arg2，on John's coming 是 arg3，论元 3（arg3）表示在哪一方面打赌，也就是打赌的内容。

当然，有时动词也可以不提打赌的内容，这时，arg3 就等于零了。例如，句子

He bets me ten dollars.（他与我打赌十元。）

在这种情况下，英语的 bet 这个动词的句法特征可用如下的规则来表示：

当不提打赌的内容时，表示为规则 1。

规则 1：

VP → V X1 X2

<V arg1> = X1

<V arg2> = X2

<V arg3> = 0

当提到打赌的内容时，表示为规则 2。

规则 2：

VP → V X1 X2 X3

<V arg1> = X1

<V arg2> = X2

<V arg3> = X3

一般说来，用 arg0, arg1, arg2, arg3 四个论元来描述英语动词已经足够了。

上述的表示方法是针对一个一个的英语动词的。英语中动词成千上万，仅像 love 这样的及物动词，常用的就有数千个，如果都是这样一个动词一个动词地来逐一进行描述，词库的容量将会变得十分庞大。为了避免这种困难局面，我们可以采用一种简便的“**宏表示法**”（Macros）。

宏表示法把动词加以分类，按类来记录动词的复杂特征。在英语的描述中，宏表示法把英语动词分为四类：

（1）不及物动词：如 die（死，凋谢）。

在句子

The flowers soon die.（花很快就凋谢了）

中，die 的 arg0 是 flowers（花），它是一个做主格主语的 NP。

这一类不及物动词的宏表示法如下：

Macro syn_iV:

<cat> = V

<arg0 cat> = NP

<arg0 case> = nom

其中，syn_iV 表示不及物动词（intransitive verb）的句法特征。

（2）及物动词：如 eat（吃）。

在句子

Tigers eat meat.（老虎吃鲜肉）

中，eat 的 arg0 是 tigers（老虎），它是一个做主格主语的 NP，eat 的 arg1 是 meat（鲜肉），它是一个做宾格宾语的 NP。由于主格主语在不及物动词的宏表示法 Macro syn_iV 中已经出现过，故不再重复写出，简写为 syn_iV 即可。这一类及物动词的宏表示法如下：

Macro syn_tV:

syn_iV

<arg1 cat> = NP

<arg1 case> = acc

其中，syn_tV 表示及物动词（transitive verb）的句法特征。

在调用 Macro syn_tV 时，应该同时激活 Macro syn_iV，也就是说，Macro syn_tV 应该与 Macro syn_iV 一块儿调用。

（3）双及物动词：如 give（给）。

在句子

We give a book to the boy.（我们给了这个男孩儿一本书）

中，give 的 arg0 是 we（我们），它是一个做主格主语的 NP，give 的 arg1 是 a book（一本书），它是一个做宾格宾语的 NP，give 的 arg2 是 to the boy，它是一个介词形式（pform）为 to 的 PP。由于主格主语在不及物动词的宏表示法 Macro syn_iV 中已经出现过，宾格宾语在及物动词的宏表示法 Macro syn_tV 中已经出现过，故不再重复写出，只简写为 syn_tV。这一类双及物动词的宏表示法如下：

Macro syn_dtV:

syn_tV

<arg2 cat> = PP

<arg2 pform> = to

其中，syn_dtV 表示双及物动词（ditransitive verb）的句法特征。

在调用 Macro syn_dtV 时，应该同时激活 Macro syn_tV，也就是说，Macro syn_dtV 应该与 Macro syn_tV 一块儿调用，而当调用 Macro syn_tV 时，又得激活 Macro syn_iV，所以，在调用 Macro syn_dtV 时，Macro syn_tV 及 Macro syn_iV 都激活了。

（4）给予动词：如 hand（递交）。

在句子

My brother hands me the hammer.（我的弟弟把锤子送给我）

中，hand 的 arg0 是 my brother（我的弟弟），它是一个做主格主语的 NP，hand 的 arg1 是 the hammer（锤子），它是一个做宾格宾语用的 NP，hand 的 arg2 是 me（我），它是另一个做宾格宾语的用的 NP，由于主格主语在不及物动词的宏表示法 Macro syn_iV 中已经出现过，第一个宾格宾语在及物动词的宏表示法 Macro syn_tV 中已经出现过，故不再重复写出，只简写为 syn_tV，这一类给予动词的宏表示法如下：

Macro syn_datV:

syn_tV

<arg2 cat> = NP

<arg2 case> = acc

其中，syn_datV 表示给予动词（dative verb）的句法特征。

在调用 Macro syn_datV 时，应该同时激活 Macro syn_tV，而激活 Macro syn_tV 时，也必得要先激活 Macro syn_iV，这样，在调用 Macro syn_datV 时，Macro syn_tV 和 Macro syn_iV 都激活了。

这种宏表示法大大地简化了词汇的句法特征的写法，它用一个简单的符号来代替一大串复杂特征。例如，用 syn_iV 这样的简单符号，就代替了<cat>=V, <arg0 cat>=NP, <arg0 case>=nom 等复杂特征。在词汇条目中，每当我们调用一个宏表示时，也就等于调用了它所代替的一大串复杂特征，我们甚至可以用一个宏表示来定义另一个宏表示，例如，用宏表示 syn_iV 来定义宏表示 syn_tV。

采用这些手段，我们可以把词汇条目表达得十分简洁。

例如，我们可以把 die（死，凋谢），elapse（消逝），eat（吃），give（给），hand（递交），love（爱，喜欢）等单词条目用宏表示法写成如下的形式：

Lexeme die:

syn_iV.

Lexeme elapse:

syn_iV.

Lexeme eat:

syn_iV.

Lexeme eat:

syn_tV.

Lexeme give:

syn_tV.

Lexeme give:

syn_dtV.

Lexeme give:

syn_datV.

Lexeme hand:

syn_dtV.

Lexeme hand:

syn_datV.

Lexeme love:

syn_tV.

有些词可以属于不同的句法类别，因而它们可以归入若干个不同的词汇条目。例如，eat 可以为不及物动词，又可为及物动词，故可归入词汇条目 syn_iV 和 syn_tV；give 可以为及物动词、双及物动词、给予动词，故可归入词汇条目 syn_tV, syn_dtV 和 syn_datV；hand 可以为双及物动词，又可以为给予动词，故可入词汇条目 syn_dtV 和 syn_datV。

宏表示大大地简化了词汇条目的写法，但在机器翻译的过程中，有必要对宏表示做出适当的解释，以适应机器翻译系统的特定要求。这种解释，叫做“宏表示的扩展”（expansion of Macro）。宏表示扩展的详略程度视自然语言处理系统的不同要求而有所不同，必要时，我们甚至可以把宏表示直接扩展为词汇条目的非循环有向图 DAG。

例如，宏表示

Lexeme give:

syn_tV.

可以扩展为如下的非循环有向图：

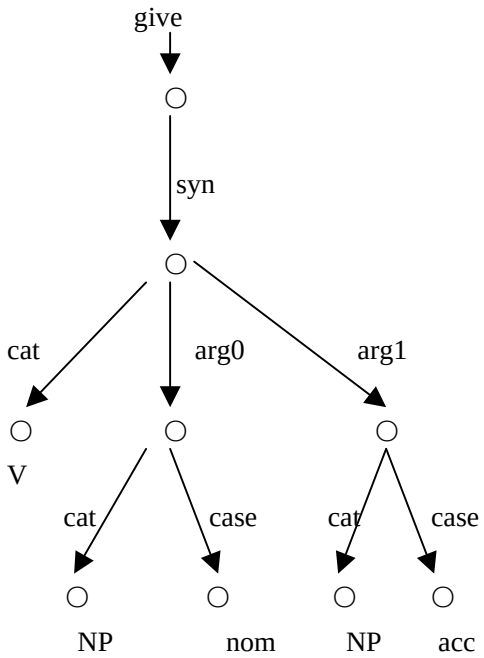

图 1　宏表示扩展为非循环有向图

当然，根据机器翻译系统的实际需要情况，我们有时只是把宏表示扩展为非循环有向图中的一部分。

把宏表示扩展之后，便可以与其他词汇单元的非循环有向图进行合一，剖析程序便可以利用词汇条目中所包含的复杂特征进行运算。

上面我们只是研究了词汇的句法信息的表示方法，事实上，词汇中还包含语义信息和词法信息，我们在词汇条目的复杂特征描述中，有必要全面地表示出词汇中所包含的各种信息，既要描述句法信息，也要描述语义信息和词法信息。

词汇的语义信息，对于动词来说，主要是它的论元信息。例如，动词 eat 可有不同的论元。在句子

We eat.（我们吃）

中，动词 eat 只有一个论元 arg0（we）；

在句子

We eat fish.（我们吃鱼）

中，动词 eat 有两个论元：arg0（we）和 arg1（fish）。因此，在语义上，我们有必要把动词 eat 分为两个：只有一个论元的 eat 记为 eat1a，具有两个论元的 eat 记为 eat2a，其中的数目字表示论元的个数，1 表示有一个论元，2 表示有两个论元。

依此推之，在句子

We give fish to John.（我们把鱼给约翰）

中的 give 有三个论元：arg0, arg1, arg2, 我们在语义上把 give 记为 give3a。

在句子

We give John fish.（我们给约翰鱼）

中的 give 也有三个论元：arg0, arg1, arg2, 但是，其中的 arg2 不带介词 to，为与 give3a 相区别，我们在语义上把这个 give 记为 give3b。

这里的 1a, 2a, 3a, 3b 等只是一种语义常数, 不同的词的语义常数不尽相同，这样，从语义常数我们就不难看出词在语义上的特性。

如果我们用宏表示来记录词汇的句法信息，用<sem>来记录词汇的语义信息，那么，die, elapse, eat, give, hand, have 等单词条目可以进一步表示如下：

```
Lexeme die:
    syn_iV
    <sem> = die1a.
Lexeme elapse:
    syn_iV
    <sem> = elapse1a.
Lexeme eat:
    syn_iV
    <sem> = eat1a.
Lexeme eat:
    syn_tV
    <sem> = eat2a.
Lexeme give:
    syn_tV
    <sem> = give2a.
Lexeme give:
    syn_dtv
    <sem> = give3a.
Lexeme give:
    syn_datV
    <sem> = give3b.
Lexeme hand:
    syn_dtV
    <sem> = hand3a.
Lexeme hand:
    syn_datV
```

<sem> = hand3b.

Lexeme love:

syn_tV

<sem> = love2a

在词汇条目中，我们还需要描述词法信息。英语的一个动词最多可以有八个不同的形式。其中一个形式是词根，其他七个形式表示不同的语法语义。

例如，英语的不规则动词 be 的八个形式如下：

root - be

form1 - am

form2 - are

form3 - is

form4 - was

form5 - were

form6 - been

form7 - being

我们用特征 root 来表示动词的词根，用特征 form1 到 form7 来表示动词的其他七个形式：form1，form2 和 form3 表示第一人称，第二人称和第三人称的现在时形式，form4 表示第一人称单数过去时形式，form5 表示第二人称单数过去时形式，form6 表示过去分词形式，form7 表示现在分词形式。不规则动词 be 的这八个形式在形态上各不相同，而且词根与其他七个形式在形态上的联系也不是一眼就可以看出来的。

英语的规则动词只有四种不同的形式，而且，它们在形态上可以从词根推出来。例如，stamp（盖章）的形式如下：

root—stamp

form1—stamp

form2—stamp

form3—stamps

form4—stamped

form5—stamped

form6—stamped

form7—stamping

为了分析上的方便，我们把规则动词的这些形式分为两个部分：一部分叫词干（stem），一部分叫词尾（ending），这样，我们就可以用宏表示 Macro mor_regV 来记录规则动词的词法信息。在宏表示 Macro mor_regV 中，mor 表示词法（morphology），regV 表示规则动词（regular verb）。

Macro mor_regV:

<mor form1 stem> = <mor root>

<mor form1 ending> = ε

<mor form2 stem> = <mor root>

<mor form2 ending> = ε

<mor form3 stem> = <mor root>

<mor form3 ending> = s

<mor form4 stem> = <mor root>

<mor form4 ending> = ed

<mor form5 stem> = <mor root>

<mor form5 ending> = ed

<mor form6 stem> = <mor root>

<mor form6 ending> = ed

<mor form7 stem> = <mor root>

<mor form7 ending> = ing

这里，mor 表示词法，stem 表示词干，ending 表示词尾，ε 表示空词尾，也就是语法中的零形式。在宏表示 Macro mor_regV 中，当词干与词尾结合成为词的各种形式时，应遵循英语正词法规则。例如，当词干 love 与词尾 ing 结合时，love 中的 e 应该抹去，结合后应该形成 loving，而不能形成 loveing。

如果我们在一个英语词条中，同时考虑句法、语义和词法的信息，并使用宏表示法，那么，英语词条可表示得十分紧凑和简洁。例如，stamp 这个词条可表示为：

Lexeme stamp:

<mor root> = stamp

mor_regV

syn_tV

<sem> = stamp2a

这种表示法中的第一行与第二行有些重复，因为词条名与词根的形式是等同的。为了表达的简洁性，我们提出如下的规定：

如果有词条

Lexeme xxx:

<mor root> = xxx

yyy

…

zzz

我们可以将其简写为

Lexeme xxx

yyy

…

zzz

这样一来，stamp 这个词条可简写为：

Lexeme stamp:

mor_regV

syn_tV

<sem> = stamp2a

根据宏表示的含义以及有关的简写规定，这个词条包含的信息可解释如下：

Lexeme stamp:

<mor root> = stamp

<mor form1 stem> = stamp

<mor form1 ending> = ε

<mor form2 stem> = stamp

<mor form2 ending> = ε

<mor form3 stem> = stamp

<mor form3 eding> = s

<mor form4 stem> = stamp

<mor form4 ending> = ed
<mor form5 stem> = stamp
<mor form5 ending> = ed
<mor form6 stem> = stamp
<mor form6 ending> = ed
<mor form7 stem> = stamp
<mor form7 ending> = ing
<syn cat> = V
<syn arg0 cat> = NP
<syn arg0 case> = nom
<syn arg1 cat> = NP
<syn arg1 case> = acc
<sem> = stamp2a

英语中的规则动词都可以用这样的方法来表示，对于 love 这样的规则动词，只须考虑英语正词法的有关规定，处理一下 love 后面的 e，做起来也不困难。对于英语中的不规则动词，则应该根据它们在形态上的特点，对词法的宏表示作适当的调整和修改。例如，eat 和 give 这两个动词，它们的单数第一人称过去时与单数第二人称过去时相同，且具有特殊的形态，eat 的特殊形态为 ate，give 的特殊形态为 gave，它们的过去分词均加词尾 en，而它们的现在时与现在分词形式则与其他规则动词一样，因此，我们可以为它们写一个宏表示 Macro mor_presV，定义如下：

Macro mor_presV:
<mor form1 stem> = <mor root>
<mor form1 ending> = ε
<mor form2 stem> = <mor root>
<mor form2 ending> = ε
<mor form3 stem> = <mor root>
<mor form3 ending> = s
<mor form4 stem> = <mor form5 stem>
<mor form4 ending> = ε
<mor form5 ending> = ε
<mor form6 stem> = <mor root>
<mor form6 ending> = en
<mor form7 stem> = <mor root>
<mor form7 ending> = ing

如果我们采用上述的简写方法，用宏表示来记录词汇的词法信息和句法信息，用<sem>来记录词汇的语义信息，那么，die, elapse, eat, give, hand, love 等单词条目可以完整而简洁地表示如下：

Lexeme die:
mor_regV
syn_iV
<sem> - die1a.
Lexeme elapse:
mor_regV
syn_iV

<sem> = elapse1a.

Lexeme eat:

mor_presV

<mor form4 stem> = ate

syn_iV

<sem> = eat1a.

Lexeme eat:

mor_presV

<mor form4 stem> = ate

syn_tV

<sem> = eat2a.

Lexeme give:

mor_presV

<mor form4 stem> = gave

syn_tV

<sem> = give2a.

Lexeme give:

mor_presV

<mor form4 stem> = gave

syn_dtV

<sem> = give3a.

Lexeme give:

mor_persV

<mor form4 stem> = gave

syn_datV

<sem> = give3b.

Lexeme hand:

mor_regV

syn_dtV

<sem> = hand3a.

Lexeme hand:

mor_regV

syn_datV

<sem> = hand3b.

Lexeme love:

mor_regV

syn_tV

<sem> = love2a.

这样一来，我们便可以十分方便地用复杂特征来描述词汇知识和表达词汇知识。一个单词经过了自动形态分析之后，词尾和词干都已经确定，再通过我们这里所述的词汇知识的复杂特征表示法，在分析词典中使用复杂特征来记录词汇知识，这必定会有效地提高机器翻译中句子自动剖析的准确性。

2　转换词典中语言信息的形式表示方法

转换词典用于从源语言到目标语言的词汇转换。词汇转换程序主要建立在查询转换词

典的基础上。这里我们首先介绍转换词典条目的结构，然后再介绍词汇转换程序的结构。

1. 转换词典条目的结构

转换词典的一个条目相当于一条规则，它除了可进行单词与单词的转换之外，还可以把源语言树形图的一个结点转换为目标语言的一个子树形图，当然，这个子树形图也可以退化为一个结点。

在 FAJRA 机器翻译系统中，我们用程序设计中的“巴库斯—瑙尔范式”（Bacus-Naur Form）来定义转换词典中的词典条目的结构，定义如下：

<条目> ::= <源语言词汇单元词符> == <三元表><后三元表>

<源语言词汇单元词符> ::= '<源语言词汇单元>'

<三元表> ::= ϕ|<条件>/<子树形图>/<赋值>/<三元表>

<后三元表> ::= <子树形图>/<赋值>

<条件> ::= <源语言特征的直接布尔条件>/$<条件过程名>

<子树形图> ::= ϕ|<子树>

<子树> ::= <恒等元>|<恒等元>(<结点表>)

<恒等元> ::= <1-8 个字母数字的名>

<结点表> ::= <子树>|<结点表><子树>

<赋值> ::= <赋值>|<赋值>;<赋值>

<赋值> ::= <恒等元>:<目标语言赋值>|<目标语言赋值>

<目标语言赋值> ::= <目标语言词汇单元词符><偏特征组合><基本赋值表>

<目标语言词汇单元词符> ::= '<目标语言词汇单元>'

<偏特征组合> ::= ϕ|,*<特征组合名>|,+<特征组合名 >

<基本赋值表> ::= ϕ|,<基本赋值>,<基本赋值表>

<基本赋值> ::= <直接赋值表达式>|<赋值过程名>

在这些巴库斯—瑙尔范式中，“::=”表示定义，它前面是定义的对象，它后面是定义的内容。“<三元表>”是包括“<条件>”、“<子树形图>”和“<赋值>”三个项目的线性表。“<后三元表>”中不包括“<条件>”，在不满足“<三元表>”中的条件时，就直接执行“<后三元表>”，因此，“<后三元表>”是在“<三元表>”之后执行的，在“<后三元表>”的定义中，只需要包括“<子树形图>”和“<赋值>”两个项目，不再需要“<条件>”这个项目。

在输入树形图的每一个结点上都有标记。包含在这个标记中的源语言词汇单元的值，就是在转换中查找词典条件的根据。

为了知道是否可以执行，算法查找第一个<三元表>，对于这个<三元表>中的“<条件>”，在输入树形图的当前结点上进行验证，看其是否相符合，如果符合就可执行。

在后三元表中不包含“<条件>”，因此，当不满足<三元表>中的条件时，就自动地转入执行<后三元表>。也就是说，查<后三元表>，计算机程序总可以得到一个解。

<三元表>中的第二项（即“<子树形图>”）在当前结点上进行替换。

如果“<子树形图>”由若干个结点构成，则它的根取源语言结点的根，根的后裔置于源语言中被替换的这个结点的后裔的左侧。

例如，如果词典条目为：

```
'ULA'   == /C1(C2,C3)/C1: 'ULCB1';
                       C2: 'ULCB2';
                       C3: 'ULCB3'.
```

输入树形图为：

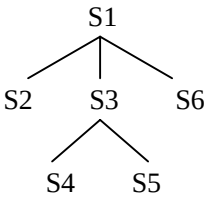

图 2　输入树形图

如果在图 2 的输入树形图中，S1 的词汇单元 UL = 'ULA', 而这个词汇单元是一个子树形图 C1(C2,C3)，则作如下转换：

输入树形图　　输出树形图

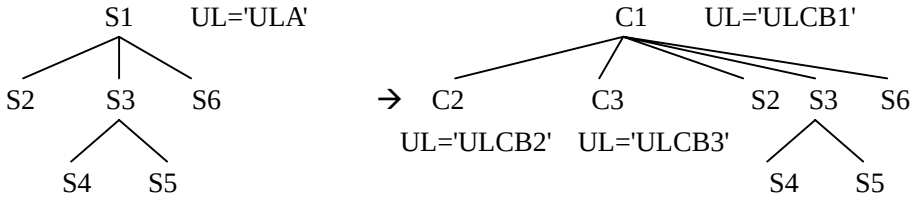

图　3　　词汇转换

根 C1 的后裔 C2 和 C3 置于 S1 的各个后裔的左侧。在新接入的子树形图的各个结点上，都应分别注明其标记 C1, C2，C3，词汇单元 UL 的词符及其特征值由三元表中的赋值部分给出，分别为：C1 的赋值为 UL='ULCB1'，C2 的赋值为 UL='ULCB2'，C3 的赋值为 UL='ULCB3'。赋值的定义为“<恒等元>:<目标语言赋值>”，C1, C2, C3 是“恒等元”，'ULCB1'，'ULCB2'和'ULCB2'是“目标语言赋值”。

2. 词汇转换程序的结构

由于在词汇转换时标记的特征既涉及到源语言，也涉及到目标语言，所以，词汇转换程序中的特征值可以包括两部分：

第一、由源语言中保持下来的旧特征值；

第二、目标语言中引入的新特征值。

词汇转换程序的外部数据分为五个部分：

第一、特征说明；

第二、本点条件特征组合；

第三、赋值特征组合；

第四、本点条件过程和赋值过程；

第五、词典。

其中，特征说明部分和词典部分是必不可少的。

下面我们来分别解释这五个部分。

（1）特征说明：

特征说明分为三种类型：互斥特征、非互斥特征、算术特征。每一个特征可以用一个特征名及其值的表来定义。

在程序的安排上，我们用“-EXC-”表示互斥特征部分，用“-NEX-”表示非互斥特征部分，用“-ARITH-”表示算术特征部分。并且规定：

① 互斥特征写在符号-EXC-之后，每个特征只能取特征值表中的一个值。

② 非互斥特征写在符号-NEX-之后，每一个特征可以取特征值表中的若干个特征值为其值。

③ 算术特征写在符号-ARITH-之后，每一个特征只能取特征值表中的一个特征值，这个值处于-(n+1)和 n 之间，其中，n 是所说明的正数的界。

注意：第一，在词汇转换前的阶段所用过的特征，如果在词汇转换阶段还想保存，那么，在词汇转换器的程序中，写为

NOM := (*)

其中，“* ”号表示 NOM 这个特征在词汇转换之前的阶段（例如，在源语言分析阶段）曾经用到的特征值。

第二，在词汇转换器中，还可以定义子特征，它们是从其他特征中分割出来的更小的特征。

例如：VARG1 := (VARS1(*), VART).

其中，VARS1(*), VART 就是 VARG1 的子特征。

（2）特征组合

特征组合（format）是由特征值组合而成的。它由特征组合名后面跟着特征值的表达式构成，各个表达式之间用逗号分开。如果特征值是常数，就不能用表达式算符“:=”，而要用等号“=”。

特征组合有两种类型：

第一、本点条件的特征组合：其特征名和特征值在前面模型的源语言的特征的集合中来取。这个特征组合说明在树形图的当前结点上的条件。

第二、赋值的特征组合：其特征名和特征值在目标语言的特征的集合中来取。这个特征组合说明赋值的情况。

例如，在我们设计的 FAJRA 多语言机器翻译系统的汉法词汇转换中，我们取如下的特征组合 PRESF 来表示法语的阴性人称代词。

PRESF 01==ELLE .** CAT-E-R, SUBR-E-PERS, GNR-E-FEM.

这是赋值的特征组合，PRESF 这个特征组合用来给法语单词“ELLE”（她）赋值，具体的赋值内容是：CAT（范畴）为 R（代词），SUBR（代词的次类）为 PERS（人称代词），GNR（性）为 FEM（阴性）。

细心的读者可以看出，我们这里使用的“特征组合”（format）与在分析词典中使用的“宏表示”（Macros）很接近，它们的原理是一样的，都可以简化特征的表示。

（3）条件过程和赋值过程

过程（procedure）由参数名后面跟着一个或多个带特征的表达式构成，过程也是一种“宏表示”（Macros）方法。

条件过程只涉及源语言的特征。

赋值过程则要涉及目标语言的特征。

在程序的安排上，我们用“-PCP-”表示本点条件过程部分，用“-PAF-”表示赋值过程部分，用“-FIN-”表示过程结束。并且规定：

① 本点条件过程写在符号“-PCP-”之后。

② 赋值过程写在符号“-PAF-”之后。

③ 符号“-FIN-”写在过程最后，表示过程部分结束。

例如，在 FAJRA 多语言机器翻译系统的汉法词汇转换中，我们使用了如下的过程：

-PCP-

PLU == NUM-E-PLU.

FRAME == FRAME-E-DEBUT.

TIME == SEM-E-TIME.

PHYS == CHAMP-E-PHYS.

这一部分是本点条件过程，用来说明词典条目中的条件，符号“-E-”相当于等号，用于说明条件。例如，PLU 表示当前结点的数（NUM）为“复数”（PLU），FRAME 表示当

前结点为“短语的开头”（DEBUT），TIME 表示当前结点的语义（SEM）为“时间”（TIME），PHYS 表示当前结点的学科领域（CHAMP）为“PHYS”（物理学）。

-PAF-

MODAL == CAT:=V, TYPV:=MODAL.
PP == K:=PP.
COP == CAT:=V, TYPV:=COP.
ADJA == CAT:=A, SUBA:=ADJ, SEM:=ANIM.
CNFP == CAT:=N, SUBN:=CN, GNR:=FEM, NUM:=PLU.
CNMP == CAT:=N, SUBN:=CN, GNR:=MAS, NUM:=PLU.
CNFS == CAT:=N, SUBN:=CN, GNR:=FEM, NUM:=SIN.

这一部分是赋值过程，用于在转换词典中给目标语言法语赋值。例如，MODAL 表示给当前结点的赋值是：“CAT:=V, TYPV:=MODAL”，也就是范畴（CAT）为“单词”（V），动词次类（TYPV）为“情态动词”（MODAL）。赋值符号不用“-E-”，而用“:=”。PP 表示词组类型（K）的赋值是“介词短语”（PP）；COP 表示范畴（CAT）的赋值是动词（V），动词次类（TYPV）的赋值是系动词（COP）；ADJA 表示范畴（CAT）的赋值是修饰词（A），修饰词次类（SUBA）的赋值是形容词（ADJ），语义（SEM）的赋值是“有生命的”（ANIM）；CNFP 表示范畴（CAT）的赋值是名词（N），名词次类（SUBN）的赋值是普通名词（CN），语法性（GNR）的赋值的阴性（FEM），数（NUM）的赋值是复数（PLU）；CNMP 表示范畴（CAT）的赋值是名词（N），名词次类（SUBN）的赋值是普通名词（CN），语法性（GNR）的赋值的阳性（MAS），数（NUM）的赋值是复数（PLU）；CNFS 表示范畴（CAT）的赋值是名词（N），名词次类（SUBN）的赋值是普通名词（CN），语法性（GNR）的赋值的阴性（FEM），数（NUM）的赋值是单数（SIN）。

容易看出，由于在本点条件过程和赋值过程使用了类似于我们在分析词典中的“宏表示”（Macros）的方法，大大地简化了过程的写法。

过程部分结束时使用符号“-FIN-”，写在过程的最后。

（4）词典

词典是词条的系列。每个词条包括如下的内容：

——源语言词汇单元名,写在两个引号 '' 之中；

——分隔符“==”；

这一部分没有特殊的限制，但应该注意的是，语言的全部词汇单元应该写在每行的开头。

——三元表，又包含如下三项内容：

① 条件：本点条件表达式或调用条件过程，条件过程名的前面要加符号“$”，如 $SIN。

② 当前结点转换成的子树形图，子树形图用括号表达式，写法如前，如果子树形图退化为一个结点，则这部分内容可以不写。

③ 子树形图每个结点上的赋值。顺序为：

结点名；

符号“:”。

在结点上赋值的目标语言词汇单元名写在两个引号 ' ' 中；

目标语言特征的赋值表，它可包含：

① 带有前缀的特征组合名，即偏特征组合，它要位于最前；

② 赋值表达式，即不用特征组合来赋值，而直接用特征来赋值；

③ 调用赋值过程，调用时，其顺序不受限制。

在赋值时，偏特征组合名的前面要加前缀“+”或“*”，赋值过程名前面要加前缀“$”。这三个前缀的功能是不同的。

所有的偏特征组合，都由词汇转换程序中重新定义的特征值所组成，不能由前面阶段的旧特征来赋值。

如果在偏特征组合名的前面加前缀“*”，则用该偏特征组合所赋值的词汇单元所带的特征值中，不能保存旧特征值，前面阶段的旧特征值均被抹去。

如果在偏特征组合名的前面加前缀“+”，则用该偏特征组合来赋值的词汇单元所带的特征值中，不仅能保存该偏特征组合所赋的新特征值，而且还能保存前面阶段所得到的旧特征值。

在赋值过程名的前面，必须加符号“$”作为前缀，该赋值过程所赋值的词汇单元所带的特征值中，不仅可包含用该赋值过程来赋值的旧特征值和新特征值，而且还能保存前面阶段所得到的旧特征的全部值。列表比较如下：

	能否用新特征赋值？	能否用旧特征赋值？	能否保存旧值？
$	能	能	能
+	能	不能	能
*	能	不能	不能

表 1 前缀的功能比较

例如，在 FAJRA 多语言机器翻译系统的汉法词汇转换的词典中，我们有如下的词条：

'地震' == /0(1,2(3,4))/0: 'UL0';
1: 'SECOUSSE', $CNFEP;
2: '*PP', K:=PP;
3: 'DE', *PREP;
4: 'SOL', +CNMAS.

'UL0'表示零词汇单元，我们用它来引导一个子树形图。这个词条可以把汉语单词“地震”转换为法语的短语“secousse de sol”。图示如下：

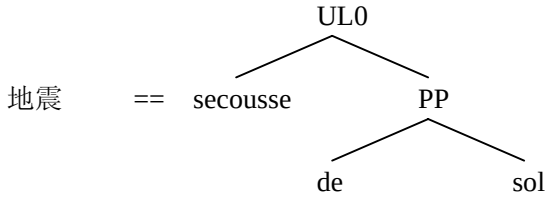

图 4

对于法语短语“secousse de sol”中的“DE”，我们使用了偏特征组合“*PREP”，这是由于“地震”在汉语中是个名词（或动词），它一般可保存特征 FS （句法功能）的一个值，这个值就是 GOV（中心词），但在法语中，“DE”是一个介词，显然不能用 GOV 来赋值，因而我们在 PREP 这个特征组合之前加前缀“*”，以便抹去 GOV 这个特征值。

在 FAJRA 多语言机器翻译系统的汉法转换词典中，还有如下的例子：

'龟甲' == /0(1,2(3,4))/0: 'UL0';
1: 'ECAILLE', $CNFP;
2: '*PP', *ABS, K:=PP;
3: 'DE', *PREP;
4: 'TORTUE', $CNFP.

图示如下：

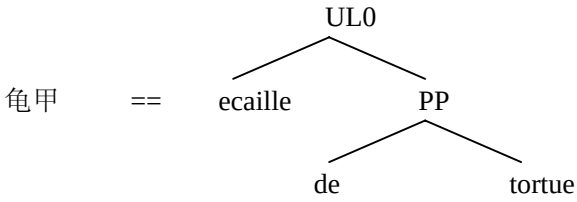

图 5

这个词条可以把汉语单词“龟甲”转换成法语短语“ecaille de tortue”，其中，对于法语介词“DE”，我们在 PREP 前面也加了前缀“*”。其理由与“地震”词条中的相同。

'汉字' == /0(1,2)/0: 'UL0', +CNMAS;
1: 'CHINOIS', *ADJ;
2: 'CARACTERE', +CNMAS.

图示如下：

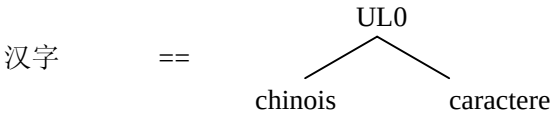

图 6

这个词条可以把汉语单词“汉字”转换成法语短语“chinois caractere”，其中，对于法语形容词“CHINOIS”，我们也在 ADJ 前面加了前缀“*”，因为在“chinois caractere”中，“chinois”不是中心词，不能保持原来在汉语中的信息“ GOV”。

'袈裟' == /0(1,2)/0: 'UL0', +CNMAS;
1: 'SACERDOTAL', *ADJ;
2: 'VETEMENT', +CNMAS.

图示如下：

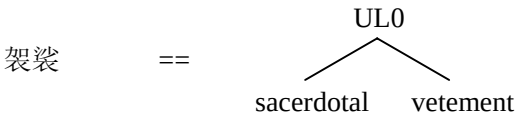

图 7

这个词条可以把汉语单词“袈裟”转换成法语短语“sacerdotal vetement”，其中，对于法语形容词“SACERDOTAL”，我们也在 ADJ 前面加了前缀“*”，因为在“sacerdotal vetement”中，“sacerdotal”不是中心词，不能保持原来在汉语中的信息“ GOV”。

'日全食' == /0(1,5(6),2(3,4))/0: 'UL0';
1: 'ECLIPSE', +CNFEM;
2: '*PP', *ABS, K:=PP;
3: 'DE', *PREP;
4: 'SOLEIL', +PNMAS;
5: '*AP', K:=AP;
6: "TOTAL', +ADJ.

图示如下：

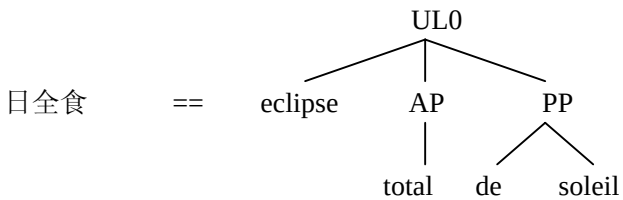

图 8

这个词条可以把汉语单词“日全食”转换为法语短语“eclipse total de soleil”，其中，对于法语介词“DE”，我们也在 PREP 前面加了前缀“*”，因为在“eclipse total de soleil”中，“de”不是中心词，不能保持原来在汉语中的信息“ GOV”。

3 生成词典中语言信息的形式表示方法

FAJRA 机器翻译系统中的形态生成程序由两个转录机所组成：一个是“树—链转录机”，它的任务是生成目标语言树形图的前沿，也就是把经过目标语言结构生成所得到的目标语言树形图的叶子中的词汇单元及其有关的信息取出来，把树形图改造成一个由其叶子组成的线性符号链；另一个是“链—链转录机”，它的任务是把带有形态特征标记的线性符号链转换为具有屈折形式并符合目标语言正词法规则要求的文字链，即机器翻译的译文。

形态生成程序的数据由两部分组成：

第一、特征说明，特征组合，条件过程。

第二、词典：第一部词典必须以词汇单元为词条，其他各部词典以特征值作为词条。

1、形态生成程序的外部数据

（1）特征说明：分为互斥特征和非互斥特征两部分。

对于前面阶段要保持的特征，写为

NOM := (*).

在法语形态生成中，我们提出了互斥特征 FLEX 来表示变形词的词形变化系统（inflexion paradigm），又提出非互斥特征 ELIS 来表示法语中的连音变读类别（liaison）。

ELIS 表示的法语中的连音变读是法语的一种特殊的语音现象。如

de + les →des

de + le → du

à + les → aux

à + le → au.

在输出形态生成的结果时，要处理这样的连音变读，才能得到符合法语正词法要求的译文。

（2）特征组合：与转换词典中的特征组合相同。

例如，法语形态生成的特征组合，有

ELIS1 01==A .** ELIS-E-ELIS1.

ELIS13 01== .** ELIS-E-ELIS1-∪-ELIS3.

特征组合 ELIS1 表示连音变读类别为 ELIS1，特征组合 ELIS13 表示连音变读类别为 ELIS1 和 ELIS3 的“并”，也就是说，可以为 ELIS1，也可以为 ELIS3。正因为这样，我们把连音变读作为一种非互斥特征来处理。非互斥特征可以进行“并”运算。

（3）条件过程：

条件过程的写法如下：

条件名 01== …….

条件名 01== …….

.

.

.

条件名 01== …….

条件过程的右部与特征组合的右部相似，但可包含如下的逻辑算子：-I-（交），-∪-（并），-E-（等于），-NE-（不等于），-DANS-（在其中），-NDANS-（不在其中），-INC-（包含），-NINC-（不包含），-N-（否），-ET-（与），-OU-（或）.

条件过程中，不能容许多层括号，应将括号展开，可定义如下：

无括号的条件过程右部 ::= [-N-]表达式 3 { -ET- / -OU- } [-N-]表达式 3

表达式 3 ::= 表达式 1 { -E- / -NE- / -DANS- / -NDANS- / -INC- / -NINC- } 表达式 2

表达式 1 ::= 含有-I-和-U-的特征表达式

表达式 2 ::= 特征值表达式

例如：在法语形态生成中的条件过程有：

AM 01== SUBA-E-ADV-ET-DRV-E-AM.

APLUF 01== SUBA-E-ADJ-ET-NUM-E-PLU-ET-

APLUF 02 GNR-E-FEM-OU-SUBA-E-ORD-ET-

APLUF 03 NUM-E-PLU-ET-GNR-E-FEM.

在条件名 AM 的右部“SUBA-E-ADV-ET-DRV-E-AM” 中，“SUBA-E-ADV'”是表达式 3，它表示修饰词次类（SUBA）是副词（ADV），“DRV-E-AM”也是表达式 3，它表示这个副词的派生（DRV）来自形容词（AM），这两个表达式 3 之间用-ET-相连接。在表达式 3 “SUBA-E-ADV” 中，SUBA 是表达式 1，它是一个特征表达式，ADV 是表达式 2，它是一个特征值表达式，表达式 1“SUBA”和表达式 2“ADV”之间用-E-相连接。

条件名 APLUF 的右部太长，分写为三行，第一行标以 01，第二行标以 02，第三行标以 03。这个条件过程表示复数阴性形容词（SUBA-E-ADJ-ET-NUM-E-PLU-ET-GNR-E-FEM）或者复数阴性次第数词（SUBA-E-ORD-ET-NUM-E-PLU-ET-GNR-E-FEM，ORD 表示“次第数词”，它也属于修饰词的一个次类），它们之间用逻辑算子“-OU-”（“或”）相连接。

2. 生成词典类型和结构：

形态生成词典可以分为词汇单元词典和特征值词典两种类型。词汇单元词典是关于词汇单元 UL 的,记为 DICG1（生成词典 1），它是形态生成程序必不可少的词典，特征值词典是关于特征值的，记为 DICG2（生成词典 2）。

词典条目的结构如下：

词汇单元名或特征值名 == [条件/赋值/链,]

== /赋值/链.

其中，条件用条件过程名，赋值用特征组合名，链是任何字构成的链。

例如，关于词汇单元的法语形态生成词典如下：

DICG1（生成词典 1）

ANALYSER ==PRE /FLEXPRV1/'ANALYS,

==IM /FLEXIMV1/'ANALYS,

==PS /FLEXPSV1/'ANALYS,

==FT /FLEXFTV1/'ANALYS,

==PAPA /FLEXPP1 /'ANALYSÉ,

== /FLEXV1 /'ANALYS.

在条件部分，PRE 表示“现在时”，IM 表示“未完成过去时”，PS 表示“过去时”，FT

表示将来时，PARA 表示“过去分词”。在赋值部分，FLEXPRV1，FLEXIMV1，FLEXPSV1，FLEXFTV1，FLEXPP1，FLEXV1 都是特征值，它们都表示法语动词的不同时态特征。在链部分是“'ANALYS”，表示具体的词汇单元，'ANALYS 前面的“'”表示这个词汇单元是以元音开头的，要做特殊的“连音变读”处理。该条目的最后一行没有写明条件，表示“缺省值”，作为这个词汇条目的出口，这样，一旦条件都不能满足时，仍然可以进行处理。

在 DICG1 中，以词汇单元为词典条目，但是，在 DICG2 中，则应以特征值为词典条目。

例如，关于特征值的法语形态生成词典如下：

DICG2（生成词典 2）

特征值条目 FLEXPRV1：

```
FLEXPRV1    ==SIN1      /       /E,
            ==SIN2      /       /ES,
            ==SIN3      /       /E,
            ==PLU1      /       /ONS,
            ==PLU2      /       /EZ,
            ==PLU3      /       /ENT.
```

注意，在特征值的条目中，只有“条件”部分和“链”部分，没有“赋值”部分，因为既然已经生成了“链”，就没有必要再继续赋值了。

特征值 FLEXPRV1 可以生成法语动词现在时词尾。“条件”中，SIN1 表示单数第一人称，SIN2 表示单数第二人称，SIN3 表示单数第三人称，PLU1 表示复数第一人称，PLU2 表示复数第二人称，PLU3 表示复数第三人称。这样，由现在时词干 analys 可以生成：

analys + e → analyse （单数第一人称现在时）
analys + es → analyses （单数第二人称现在时）
analys + e → analyse （单数第三人称现在时）
analys + ons → analysons （复数第一人称现在时）
analys + ez → analysez （复数第二人称现在时）
analys + ent → analysent （复数第三人称现在时）

特征值条目 FLEXIMV1：

```
FLEXIMV1    ==SIN1      /       /AIS,
            ==SIN3      /       /AIT,
            ==PLU1      /       /IONS,
            ==PLU2      /       /IEZ,
            ==PLU3      /       /AIENT.
```

特征值 FLEXIMV1 可生成法语动词未完成过去时词尾，由未完成过去时词干 analys 可以生成：

analys + ais → analysais （单数第一人称未完成过去时）
analys + ait → analysait （单数第三人称未完成过去时）
analys + ions → analysions （复数第一人称未完成过去时）
analys + iez → analysiez （复数第二人称未完成过去时）
analys + aient → analysaient （复数第三人称未完成过去时）

特征值条目 FLEXPSV1：

```
FLEXPSV1    ==SIN1      /       /AI,
            ==SIN3      /       /A,
            ==PLU1      /       /AMES,
```

==PLU2 / /ATES,
==PLU3 / /ERENT.

特征值 FLEXPSV1 可生成法语动词过去时词尾，由过去时词干 analys 可以生成：

analys + ai → analysai （单数第一人称过去时）
analys + a → analysa （单数第三人称过去时）
analys + ames → analysames （复数第一人称过去时）
analys + ates → analysates （复数第二人称过去时）
analys + erent → analyserent （复数第三人称过去时）

特征值条目 FLEXFTV1：

FLEXFTV1 ==SIN1 / /ERAI,
==SIN3 / /ERA,
==PLU1 / /ERONS,
==PLU2 / /EREZ,
==PLU3 / /ERONT.

特征值 FLEXFTV1 可生成法语动词将来时词尾，由将来时词干 analys 可以生成：

analys + erai → analyserai （单数第一人称将来时）
analys + era → analysera （单数第三人称将来时）
analys + erons → analyserons （复数第一人称将来时）
analys + erez → analyserez （复数第二人称将来时）
analys + eront → analyseront （复数第三人称将来时）

特征值条目 FLEXPP1：

FLEXPP1 ==PARASM / /,
==PARASF / /E,
==PARAPM / /S,
==PARAPF / /ES.

特征值 FLEXPP1 可生成法语动词过去分词，“条件”中，PARASM 表示阳性单数过去分词，PARASF 表示阴性单数过去分词，PARAPM 表示阳性复数过去分词，PARAPF 表示阴性复数过去分词。这样，由过去分词词干 analysé 可以生成

analysé + ϕ → analysé （阳性单数过去分词）analysé + e → analysée （阴性单数过去分词）

analysé + s → analysés （阳性复数过去分词）
analysé + es → analysées （阴性复数过去分词）

特征值条目 FLEXV1：

FLEXV1 ==PAPR / /ANT,
== / /ER.

特征值 FLEXV1 用于其他情况，这里主要有两种情况，一种是现在分词，“条件”中写为 PAPR，一种情况是动词不定式，因为动词不定式就是动词的原形，“条件”中写为“空”，意味着没有条件，也就是“缺省值”（default），这样，由词干 analys 可以生成：

analys + ant→ analysant （现在分词）
analys + er → analyser （不定式）

由于这个条目的最后一行是缺省值，所以，在各种条件都不能满足时，仍然可以生成动词不定式，这样，就保证了在任何情况下，不论条件是否满足，都可以得到一个生成的结果。

依靠这样的生成词典，便可以把生成系统生成的目标语言单词，进行形态变化，最后得到机器翻译的译文。这样，整个的机器翻译过程就完成了。

机器翻译词典中语言信息的形式表示方法一直是计算语言学研究中的一个薄弱环节，在汗牛充栋的计算语言学文献中罕有涉及，希望本文能够引起学界的注意，产生出抛砖引玉的效果。

参考文献

[1]冯志伟，数理语言学[M]，知识出版社，上海，1985 年。
[2]冯志伟，自然语言机器翻译新论[M]，语文出版社，北京，1994 年。
[3]冯志伟，法-汉机器翻译 FCAT 系统[J]，情报科学，哈尔滨，1987 年，第 4 期。
[4]冯志伟，德-汉机器翻译 GCAT 系统的设计原理和方法[J]，中文信息学报，北京，1988 年，第 3 期。
[5]冯志伟，机器翻译专用软件[A]，语言和计算机[C]，第 3 辑， 中国社会科学出版社，北京，1987 年。
[6] Whitelock, P. & Kilby, K. Linguistic and Computational Technique in Machine Translation System Design [M], UCL Press, 1995.
[7] Trujillo, A. Translation Engines: Technique for Machine Translation [M], Springer, 1999.