TR-IT-0311

TDMT 解析変換知識作成の手引 (日韓版)

山本 和英　　小谷 昌彦 †

1999 年 8 月

Abstract

TDMT 用解析変換データ (日韓) 作成を目的とする、解析変換知識の構成とその仕様について解説を行ない、また作業手順などについて説明する。

エイ・ティ・アール音声翻訳通信研究所
ATR Interpreting Telecommunications Research Laboratories

---

† 株式会社コングレ

目次

第 1 章

TDMT 用データの構成と仕様

TDMT 用解析変換知識データの構成とその仕様を、以下の章ごとに説明する。

各種データの書式に関する注意事項

- [ ] ... 省略可能を示す。

  例 (((TARGET-PATTERN) ([GENERATION-INFO]))

- 英小文字 ... そのまま記述することを示す。
- 英大文字 ... 該当する内容を記述することを示す。

  例 (define-transfer-dic :j-k t
  ((POS JAPANESE-STRING) TARGET)
)

1.1 翻訳訓練対象テキスト

1.1.1 バイリンガル会話

1. 目的：旅行会話をトピックとし、通訳を介して収録したバイリンガル模擬会話。翻訳訓練を行なう際の訓練文として選ばれ(システム管理者が /usr/local/TDMT/tdmt-multi-dev/build/text-definitions.lispで指定)、解析変換知識作成者は韓国語訳の付与されていない一部の会話を除き対訳を考える上で参考にすることができる。

2. 管理者：データベース管理者およびシステム管理者

3. 格納場所：
   /DB/LDB/JK/JKTEXT/{C9410, C9502, C9604, S9502}/*.JKTEXT
   /DB/SLDB/LNG/EJTEXT+/*.EJTEXT
   /DB/LDB/JE/EJTEXT+/{DEN94B, SOBA93, SOBA94A}/*.EJTEXT

4. 例： /C9502/CT340061.JKTEXT

   K| [예] 룸 서비습니다.
   J| [はい] ルームサービスでございます。

   J| [あ] 朝食のルームサービスをお願いします。
   K| 아침 식사 (룰) 룸 서비스를 부탁합니다.
   J| あしたの朝八時までに持ってきてください。
   K| 내일 아침 여덟 시까지 좀 갖다 주세요.

   K| 예, 내일 아침 여덟 시요?
   J| はい、あすの朝八時でございますね。
   K| 메뉴는 정하셨습니까?
   J| メニューはお決めに <n> なられましたか。

   J| アメリカンブレックファーストでお願いします。
   K| [에] 미국식 아침으루 부탁합니다.

   K| 예, 미국식 아침이시죠?
   J| はい、アメリカンブレックファーストでございますね。
   K| 그럼 토스트루 하시겠습니까, 롤 빵으로 하시겠습니까?
   J| ではトーストになさいますか、ロールパンになさいますか。

   J| トーストにしてください。
   K| 토스트루 하겠습니다.
   :

1.1.2 基本表現集

1. 目的：広範囲な旅行会話を網羅するために作成された、対訳付き基本表現集。以下、1.1.1 バイリンガル会話に同じ。

2. 管理者：データベース管理者およびシステム管理者

3. 格納場所： /DB/LDB/JEK/JEKTEXT/{S9503, S9510, C9608}/*.JEKTEXT}

4. 例： /S9503/A4JP.JEKTEXT

```
J|XX|AU4JP001| 今晩のホテルを予約したいのですが。
E|XX|AU4JP001|I would like to make a reservation for tonight.
K|XX|AU4JP001| 오늘 밤 호텔을 예약하고 싶은데요.
J|XX|AU4JP002| 何名さまですか。
E|XX|AU4JP002|For how many people?
K|XX|AU4JP002| 몇 분이십니까?
                :
```

＊C9608 は日英対訳、S9503, S9510 は日英韓対訳。詳細は[古瀬 95]を参照のこと。

1.2 日本語形態素タギングデータ

1. 目的：文を単語単位に分解した形態素に対して、品詞や活用形などの情報を付与したデータ。翻訳訓練の際は形態素解析を行なわず、あらかじめ解析済みのこのタギングデータをもとに各解析変換知識を作成する。タギングデータは、訓練対象テキストそれぞれに対して、システム管理者が /usr/local/TDMT/tdmt-multi-dev/build/text-definitions.lispにて指定する。

2. 管理者：タギングデータ管理者およびシステム管理者

3. 格納場所：
   /DB/LDB/TDMT/JMOR/LDB-JK/{C9410, C9502, C9604, S9502}/*.JMOR
   /DB/LDB/TDMT/JMOR/LDB-JEK/{C9608, S9503, S9510}/*.JMOR
   /DB/LDB/TDMT/EMOR/LDB-JE/{DEN94B, FURU94B, SOBA93, SOBA94A}/*.JMOR
   /DB/LDB/TDMT/JMOR/SLDB/*.JMOR

4. 書式：発話 ID| 発声 ID| 文節 ID| 単語 ID| 表記形 | 読み | 正規形 | 品詞 | 活用型 | 活用形 | 音便 |

5. 例： /C9502/AF430011.JMOR

```
10||||通訳者：|||||||
10|0010|10|10|［||［|記号||||
10|0010|10|20|はい|ハイ|はい|間投詞||||
10|0010|10|30|］||］|記号||||
10|0010|20|40|ルームサービス|ルームサービス|ルームサービス|サ変名詞||||
10|0010|20|50|で|デ|だ|判定詞|形容動詞|連用||
10|0010|20|60|ございます|ゴザイマス|ございます|補助動詞|特殊サ|基本||
10|0010|20|70|。||。|記号||||
20||||申込者：|||||||
20|0020|30|80|［||［|記号||||
20|0020|30|90|あ|ア|あ|間投詞||||
20|0020|30|100|］||］|記号||||
20|0020|40|110|朝食|チョウショク|朝食|普通名詞||||
20|0020|40|120|の|ノ|の|連体助詞||||
20|0020|50|130|ルームサービス|ルームサービス|ルームサービス|サ変名詞||||
20|0020|50|140|を|ヲ|を|格助詞||||
20|0020|60|150|お|オ|お|接頭辞||||
20|0020|60|160|願い|ネガイ|願う|本動詞|五段ワ|連用||
20|0020|60|170|し|シ|する|補助動詞|サ変|連用||
20|0020|60|180|ます|マス|ます|助動詞|特殊サ|基本||
20|0020|60|190|。||。|記号||||
                :
```

＊詳細は[溝口 98]を参照のこと。

1.3 辞書

1.3.1 日本語意味コード辞書

1. 目的：角川書店「類語新辞典」の三桁の数字をもとに、形態素ごとに意味コードを付与した辞書。システムが意味距離計算により、類似用例を検索する際に参照する。また、翻訳訓練の際に意味距離計算を行ないながら知識を作成することもできる(「2.2.1 解析変換知識作成手順」参照)。
2. 管理者：日本語意味コード辞書管理者
3. 格納場所：/usr/local/TDMT/tdmt-multi-dev/j-morph/dic/jma-atr-sem-code.text
4. 書式：(REG-EXP POS SEM-CODE)
   - REG-EXP ... 日本語形態素の正規形
   - POS ... 日本語品詞シンボル
   - SEM-CODE ... 意味コード(複数指定可)
5. 例：

```
                :
("扱い" 普通名詞 "382")
("扱う" 本動詞 "380" "382" "489" "787")
("宛" 普通名詞 "109c")
("宛先" 普通名詞 "109c")
("宛名" 普通名詞 "822b")
("安い" 形容詞 "171a" "693")
("安まる" 本動詞 "405")
("安楽寺" 普通名詞 "727" "940")
("安心" サ変形容名詞 "494" "694")
("安心する" 本動詞 "494" "694")
                :
```

＊詳細は「日本語意味コード辞書作成の手引」を参照のこと。

1.3.2 日韓変換辞書

1. 目的：日本語形態素に対応する韓国語訳を記述する辞書。一語につき、一対訳が登録できる。原言語に対しては (POS JAPANESE-STRING) が一単位となるが、目的言語 (TARGET) に対しては生成情報(品詞) を伴った形態素列を記述する。さらに、その形態素列をいくつかまとめて生成情報を付けることもできる。原言語の品詞は、[溝口 98]に従うものとする。ここで記述された対訳は、変換知識のパタン中で別の対訳に変更することができる。(詳細は「1.5.7 ローカル辞書」参照)。

2. 管理者：解析変換知識作成者

3. 格納場所： /usr/local/TDMT/tdmt-multi-dev/transfer/j-k/dic/japanese-to-korean.lisp

4. 書式：

```
(define-transfer-dic :j-k t
  ((POS JAPANESE-STRING) TARGET)
 )
```

   - define-transfer-dic ... 変換辞書の定義を宣言
   - :j-k ... 翻訳の方向。この場合は日韓翻訳を示す。
   - t ... 辞書データをロードする際にハッシュテーブルを作り直す。
   - POS ... 日本語品詞シンボル
   - JAPANESE-STRING ... 日本語文字列
   - TARGET ... 目的言語の対訳と生成情報のリスト

5. 例：

```
(define-transfer-dic :j-k t
            :
  ((本動詞 "食べる") (VVERB "먹"))
  ((本動詞 "食べれる") (VVERB "먹")(VAUX1VERB "ㄹ 수 있"))
  ((本動詞 "食べ過ぎる") (NACTV "과식")(XSUFFVERB "하"))
  ((本動詞 "食べ歩きする") (VVERB "먹")(VAUX2TEND "고")(VVERB "돌아다니"))
  ((本動詞 "食事する") (NACTV "식사")(XSUFFVERB "하"))
  ((本動詞 "伸ばす") (VVERB "늘리"))
  ((本動詞 "信用する") (NACTV "신용")(XSUFFVERB "하"))
  ((本動詞 "信頼する") (VVERB "믿"))
  ((本動詞 "寝っ転がる") (VVERB "눕"))
  ((本動詞 "寝る") (VVERB "자"))
            :
 )
```

上の例で指定した生成情報 VVERB, VAUX1VERB, NACTV, XSUFFVERB, VAUX2TEND は、TDMT 基準の韓国語品詞 (詳細は別紙「ＴＤＭＴ用韓国語形態素品詞認定基準」 株式会社コングレ 1997 を参照のこと) で、各単語の連結時生ずるわかち書きや、変化形、派生形などの生成に利用される情報である (詳細は[山本 98]を参照のこと)。

1.3.3 韓国語生成辞書

1. 目的：生成処理において、文法的に正しい韓国語を生成するために、韓国語形態素の属性情報を記述する辞書。現在の韓国語生成辞書は、形態素解析辞書から自動生成を行ない、不足分は解析変換知識作成者が手作業によって追加している。

2. 管理者：システム管理者および解析変換知識作成者

3. 格納場所：
   (自動生成版)：/usr/local/TDMT/tdmt-multi-dev/k-generation/dic/korean-generation.text
   (手作業版)：/usr/local/TDMT/tdmt-multi-dev/k-generation/dic/korean-generation-rev.text

4. 書式：(KOREAN-STRING POS (INFO-NAME ATTRIBUTION-INFO))

   - KOREAN-STRING ... 韓国語文字列
   - POS ... 韓国語品詞シンボル
   - INFO-NAME ... 属性情報の名前
   - ATTRIBUTION-LIST ... 属性のリスト

5. 例：

```
("ㄴ 듯하" VAUX1ADJV (change 여 - 불규칙))
("ㄷ 만하" VAUX1ADJV (change 여 - 불규칙))
("ㄷ 뻔하" VAUX1ADJV (change 여 - 불규칙))
("르지도 모르" VAUX1VERB (change 르 - 불규칙))
("가" VVERB (change 거라 - 불규칙))
("가게" NCOMM (connect 불가))
("가격" NCOMM (connect 불가))
("가격표" NCOMM (connect 불가))
("가깝" VADJV (change ㅂ - 불규칙))
("가라오케" NCOMM (connect 불가))
("가루" NCOMM (connect 불가))
("가루약" NCOMM (connect 불가))
("가방" NCOMM (connect 불가))
("가볍" VADJV (change ㅂ - 불규칙))
("가부키" NCOMM (connect 불가))
```

*詳細は[山本 98]を参照のこと。

1.4 解析知識

解析処理では、後述の変換知識による処理がしやすいように、入力文の原言語形態素列を加工する。解析知識には、replace-word, lexical-transformation, local-transformation, total-transformationの4種類が存在し、実際の処理は必要に応じて下記の順で行なわれる。

replace-word → lexical-transformation → local-transformation (→ total-transformation)

各種解析知識の詳細は、以下の章ごとに説明する。

1.4.1 replace-word

1. 目的：タギングデータおよび形態素解析結果の表記の揺れを統一し、変換知識のパタン数および用例数を削減するための正規形置換情報。日本語を原言語とする翻訳処理に共通の知識であるため、追加や修正を行なう際は、関係者で協議の上、決定する。

2. 管理者：正規形置換情報管理者

3. 格納場所： /usr/local/TDMT/tdmt-multi-dev/j-morph/dic/replace-word-info.lisp

4. 書式：((POS REG-EXP) (REVISED-REG-EXP))

   - POS ... 日本語品詞シンボル
   - REG-EXP ... 日本語正規形
   - REVISED-REG-EXP ... 置換後の正規形

5. 例：

```
                :
((サ変名詞 "素泊り") ("素泊まり"))
((サ変名詞 "手続") ("手続き")
((普通名詞 "サンドウィッチ") ("サンドイッチ"))
((普通名詞 "シャンペン") ("シャンパン"))
((本動詞 "くりかえす") ("繰り返す"))
((本動詞 "したがう") ("従う"))
((補助動詞 "出来る") ("できる"))
((補助動詞 "下さる") ("くださる"))
                :
```

1.4.2 lexical-trnasformation

1. 目的：変換処理で形態素を扱いやすくするために、形態素情報の修正(品詞、正規形、表層形 等の修正・変更および形態素の合成・分割など)を行なう知識。

2. 管理者：解析変換知識作成者

3. 格納場所： /usr/local/TDMT/tdmt-multi-dev/transfer/j-k/a-data/lexical-*.lisp

4. 書式：
   以下のようにlexical-transformationは、矢印(=>)の左辺が形態素パタンの条件式、右辺が処理の定義式という構成になっており、それらは1つのルールの中でリスト構造をとる。

```
(define-lexical-transformation RULE-NAME :j-k PRIORITY
  (
   ((MORPH-INFO-LIST-1) (MORPH-INFO-LIST-2) ... (MORPH-INFO-LIST-N))
   =>
   (lex ((ARGUMENT) REVISED-MORPH-INFO-LIST))
   )
   .........
  )
```

   - define-lexical-transformation ... 解析知識の定義を宣言
   - RULE-NAME ... 任意のルール名。ただし、解析知識内でユニークとする。
   - :j-k ... 翻訳の方向。この場合は日韓翻訳を示す。
   - PRIORITY ... 処理の実行優先番号
   - MORPH-INFO-LIST-1, MORPH-INFO-LIST-2,...MORPH-INFO-LIST-N ... 形態素情報のリスト(記述要素=:word(表層形), :reg-exp(正規形), :pos(品詞), :conj-form(活用形), :pron(読み), :start(発声の開始), :end(発声の終了))
   - lex ... 形態素情報修正の宣言
   - ARGUMENT(引数) ... 左辺の形態素列の何番目かを示す番号と任意の文字列の並び
   - REVISED-MORPH-INFO-LIST ... 修正後の形態素情報のリスト(記述要素=:word(表層形), :reg-exp(正規形), :pos(品詞), :conj-form(活用形))。ただし、:wordと:reg-expの値は文字列で指定するか、引数の形態素列の番号で指定する。:posと:conj-formの値はシンボルで指定するか、引数の形態素列の番号で指定する。引数の番号で指定した場合は、該当する形態素が持つ値をコピーする。:all-copyは該当する形態素が持つ:pos, :conj-form, :pronの値をコピーする。:all-copyと:pos, :conj-formの両方を指定した場合は、後の指定が優先される。後述するterminal rule用に合成する場合は、:compoundの指定をこのREVISED-MORPH-INFO-LISTに入れる。

右辺の処理定義式は、普通はlexで宣言するが、左辺で指定した形態素列に修正を加えない形態素が含まれる場合や、形態素の分割のように新しい形態素を加える場合などは、以下の形態素定義式define-morphで形態素情報を定義して呼び出す。また、引数の形態素列の番号を指定する際に、異なるフィールドの情報を用いる場合には、文字列定義式define-stringで文字列としての定義を行なってから呼び出す。現在、日韓翻訳でdefine-stringは使用していない。

```
(define-morph @RULE-NAME :j-k (ARGUMENT)
  REVISED-MORPH-INFO-LIST)
```

- define-morph ... 形態素の定義を宣言
- define-string ... 文字列の定義を宣言
- @RULE-NAME ... 任意の定義式名。形態素の定義と文字列の定義を区別するため、先頭に @ か $ が必要。解析知識内でユニークとする。
- :j-k ... 翻訳の方向。この場合は日韓翻訳を示す。
- ARGUMENT(引数) ... 左辺の形態素列の何番目かを示す番号と任意の文字列の並び
- REVISED-MORPH-INFO-LIST ... 修正後の形態素情報のリスト (記述要素 =:word(表層形), :reg-exp(正規形), :pos(品詞), :conj-form(活用形))。記述方法は上に同じ。
- STRING-INFO ... 定義に用いる文字列情報 (記述要素 =:word(表層形), :reg-exp(正規形))。記述方法は上に同じ。

5. 例 :

*一般的な正規形の変更

```
(define-lexical-transformation revised-reg-i-iru :j-k 1
  (
   (((:word . "い")(:reg-exp . "いる")))
   =>
   (lex ((1) (:all-copy . 1)(:reg-exp . "いる一")))
   )
  )
```

「いる」には、上一段活用の「居る」と五段活用の「要る・射る」などの意味があるため、区別するために、正規形を変更する。このルールでは、形態素を 1 語だけ照合し、表層形が「い」、正規形が「いる」である場合に、その形態素情報の:posと:conj-formの値をコピー (:all-copy) し、正規形のみを「いる一」に書き換える処理を行なっている。

*読み情報を利用した正規形の変更

```
(define-lexical-transformation revised-reg-hu/kaze :j-k 1
  (
   (((:pos . 普通名詞)(:word . "風")(:pron . "フウ")))
   =>
   (lex ((1) (:all-copy . 1)(:reg-exp . "ふう")))
   )
  )
```

「風」には、やり方・習慣の"식"と大気の移動運動"바람"の 2 つの意味があるため、区別するために、正規形を変更する。このルールでは、形態素を 1 語だけ照合し、品詞が「普通名詞」、表層形が「風」、読みが「フウ」である場合に、その形態素情報の:posの値をコピー (:all-copy) し、正規形のみを「ふう」に書き換える処理を行なっている。

＊一般的な形態素の合成

```
(define-lexical-transformation gousei-shukuhakukyaku :j-k 3
  (
   (((:reg-exp . "宿泊")) ((:reg-exp . "客")))
   =>
   (lex ((1 2) (:pos . 普通名詞)(:reg-exp . "宿泊客")))
   )
  )
```

「宿泊客」は、「客」の対訳”손님”の意味を持たず、二語を合成して初めて”숙박객”の対訳を決定することができる。このルールでは、入力文の形態素列を部分的に照合し、1番めの形態素の正規形が「宿泊」、2番めの形態素の正規形が「客」であることを形態素パタンの条件としている。パタンの条件が合えば、二語は「普通名詞」として合成され、正規形は「宿泊客」となる。

＊読み情報を利用した形態素の合成

```
(define-lexical-transformation gousei-machidyu :j-k 3
  (
   (((:reg-exp . "町")(:pos . 普通名詞)) ((:reg-exp . "中")(:pron . "ジュウ")(:pos . 普
通名詞)))
   =>
   (lex ((1 2) (:pos . 普通名詞)(:reg-exp . "町中＜マチジュウ＞")))
   )
  )
```

日本語形態素の読み情報を参照して、形態素を合成するルール。例の「町中」には、場所”번화가”と全体”온 동네”の2つの意味があるため、単に合成すると変換処理で曖昧性が出る。そのため、合成時に読み情報を参照して区別する。このルールでは、1番めの形態素の正規形が「町」、品詞が「普通名詞」、2番めの形態素の正規形が「中」であり、読みが「ジュウ」、品詞が「普通名詞」の場合を条件としている。条件が合えば、二語は「普通名詞」として合成され、正規形は「町中＜マチジュウ＞」となる。現在、読み情報を指定できるのは、以下の15件である。

普通名詞
「方＜ホウ＞」「方＜カタ＞」「日＜タチ＞」「中＜ジュウ＞」「中＜ナカ＞」「一＜ヒト＞」「ニ＜フタ＞」「三＜ミ＞」「風＜フウ＞」
本動詞
「入れる＜ハイレル＞」「入れ＜ハイレ＞」「入れれ＜ハイレレ＞」「入れる＜イレル＞」「入れ＜イレ＞」「入れれ＜イレレ＞」

＊形態素定義式を用いた形態素の合成

```
(define-lexical-transformation gousei-kihon_kotogadekiru :j-k 3
  (
   (((:conj-form . 基本)) ((:reg-exp . "こと")(:pos . 形式名詞)) ((:reg-exp . "が
")(:pos . 格助詞)) ((:reg-exp . "できる")(:pos . 本動詞)))
   =>
```

```
   (1 @kotogadekiru)
   )
  )

(define-morph @kotogadekiru :j-k (2 3 4)
    (:pos . 助動詞)(:reg-exp . "ことができる")(:conj-form . 4))
```

「ことができる」という表現は、変換の都合で合成する必要がある。このルールでは、1番めの形態素の活用形が「基本」形、2番めの形態素の正規形が「こと」、品詞が「形式名詞」、3番めの形態素の正規形が「が」、品詞が「格助詞」、4番めの形態素の正規形が「できる」、品詞が「本動詞」であることを条件としている。処理としては、1番めの形態素に対しては何も行なわず、2番め・3番め・4番めの形態素に対して、定義式 @kotogadekiru を呼び出して合成を行ない、合成後の品詞を「助動詞」、正規形を「ことができる」、活用形には引数で指定した4番めの形態素の値を用いることを示している。

＊形態素定義式を用いた形態素の分割

```
(define-lexical-transformation bunkatsu-tegozaimasu :j-k 4
  (
   (((:reg-exp . "てございます")(:pos . 助動詞)))
   =>
   (@teoru1 @masu1)
   )
  )

(define-morph @teoru1 :j-k (1)
   (:pos . 助動詞)(:word . "てござい")(:reg-exp . "ておる")(:conj-form . 連用))

(define-morph @masu1 :j-k (1)
   (:pos . 助動詞)(:word . "ます")(:reg-exp . "ます")(:conj-form . 1))
```

助動詞「てございます」は変換の都合で分割する必要がある。このルールでは、形態素を1語だけ照合し、正規形が「てございます」、品詞が「助動詞」であることを条件としている。処理は、分割後の1番めの形態素に対して定義式 @teoru1 を呼び出し、品詞が「助動詞」、表層形が「てござい」、正規形が「ておる」、活用形が「連用」形である形態素を定義している。2番めの形態素に対しては定義式 @masu1 を呼び出し、品詞が「助動詞」、表層形が「ます」、正規形が「ます」である形態素を定義し、活用形には引数で指定した番号の形態素(「てございます」)の値を用いることを示している。

＊terminal rule 用合成

```
(define-lexical-transformation num-yen :j-k 5
  (
   (((:pos . 数詞)) ((:word . "円")))
   =>
   (lex ((1 2) (:pos . 普通名詞)(:reg-exp . "N円")(:compound 1 2)))
   )
```

)

「一円」「二円」など、数詞を含む表現は、単に形態素を合成すると、無限にパタンを用意しなくてはならない。そのため、形態素の合成を行なうが、数量表現部分は内部処理として変換を行ないたい。このような場合にはterminal ruleと呼ばれる変換処理を用い、そのための特別な合成であることを:compoundで指定する。このルールでは、1番めの形態素の品詞が「数詞」、2番めの形態素の表層形が「円」である場合に、terminal rule用に二語を普通名詞として合成している。合成後の正規形は「N円」と定義している。

6. 優先順位 (priority) 別処理一覧

優先順位は、数字の小さい方が先に処理が行われる。優先度に揺れがある場合は、括弧内に揺れの範囲を示す。

＜優先順位 1 ＞

- タギングデータの誤り修正
- 変換の都合のための品詞変更
- 変換の都合のための表層形変更
- 数詞の terminal rule 用合成

＜優先順位 2 ＞

- 変換の都合のための正規形変更 (優先順位 1 － 3)

＜優先順位 3 ＞

- 形態素の合成 (優先順位 2 － 5)

＜優先順位 4 ＞

- 形態素の分割

＜優先順位 5 ＞

- 一般的な terminal rule 用合成 (優先順位 2 、 4 － 7)

＜優先順位 6 ＞

＜優先順位 7 ＞

1.4.3 local-transformation

1. 目的：変換処理で構造を導きやすくするために、特定の形態素間の境界を表すマーカの挿入を行なう。
2. 管理者：解析変換知識作成者
3. 格納場所： /usr/local/TDMT/tdmt-multi-dev/transfer/j-k/a-data/local-*.lisp
4. 書式：
   以下のように local-transformationは、矢印 (=>) の左辺が形態素パタンの条件式、右辺が処理の定義式という構成になっており、それらは１つのルールの中でリスト構造をとる。

```
(define-local-transformation RULE-NAME :j-k PRIORITY
  (
   ((MORPH-INFO-LIST-1) (MORPH-INFO-LIST-2) ... (MORPH-INFO-LIST-N))
   =>
   (ARGUMENT & MARKER)
   )
  .........
  )
```

   - define-local-transformation ... 解析知識のパタン定義の宣言
   - RULE-NAME ... 任意のルール名。ただし、解析知識内でユニークとする。
   - :j-k ... 翻訳の方向。この場合は日韓翻訳を示す。
   - PRIORITY ... 処理の実行優先番号
   - MORPH-INFO-LIST-1, MORPH-INFO-LIST-2,...MORPH-INFO-LIST-N ... 形態素情報のリスト (記述要素 =:word(表層形), :reg-exp(正規形), :pos(品詞), :conj-form(活用形), :pron(読み), :start(発声の開始), :end(発声の終了))
   - ARGUMENT & MARKER(引数とマーカ) ... マーカの挿入位置を示した左辺の形態素列

5. 例：

＊前後に隣接する品詞情報を示すマーカ

```
(define-local-transformation cn-v :j-k 3
  (
   (((:pos . 普通名詞)) ((:pos . 本動詞)))
   =>
   (1 <cn-v> 2)
   )
  )
```

品詞の連接情報を示す一般的なマーカを挿入するルール。内容語である「普通名詞」と「本動詞」が連接する場合には、変換処理を行なう際に境界を示すマーカが必要になる。このルールでは、１番めの形態素の品詞が「普通名詞」、２番めの形態素の品詞が「本動詞」である場合、二語の形態素の間にマーカ <cn-v> を挿入している。同種のマーカは、 <num-num>, <adjnoun-cn>, <adjnoun-sn>, <adjnoun-adj>, <adjnoun-adjnoun>, <adjnoun-v>, <adjnoun-interj>, <adjnoun-rentaiadj>, <adjnoun-npron>, <cn-sn>, <cn-sadjnoun>, <cn-pron>, <cn-adv>, <cn-pronom>,

<cn-num>, <cn-cn>, <cn-adj>, <cn-rentaishi>, <cn-conj>, <cn-adjnoun>, <cn-rentaiadj>, <cn-roman>, <cn-interj>, <fn-advb>, <fn-v>, <fn-cn>, <fn-sn>, <fn-interj>, <fn-adj>, <num-cn>, <num-pron>, <pnom-adj>, <pron-cn>, <pron-sn>, <pron-adjnoun>, <pron-advb>, <pron-v>, <pron-adj>, <pron-rentaishi>, <rentaiadj-cn>, <rentaiadj-adjnoun>, <roman-cn>, <roman-roman>, <sadjnoun-adj>, <sadjnoun-v>, <sn-interj>, <sn-cn>, <sn-pron>, <sn-v>, <sn-sn>, <sn-adjnoun>, <sn-adj>, <sn-fn>, <sn-fixp>, <sn-num> の 55 種類がある。

*統一的なマーカ

```
(define-local-transformation adv-sn :j-k 3
  (
   (((:pos . 副詞)) ((:pos . サ変名詞)))
   =>
   (1 <adv-> 2)
   )
  )
```

後続する品詞を問わずに、特定の品詞＋後続表現という形の統一的なマーカを挿入するルール。このルールでは、1番めの形態素の品詞が「副詞」、2番めの形態素の品詞が「サ変名詞」である場合にマーカを挿入しているが、後接する形態素の品詞を特定しなくても原言語構文解析が行なえる場合は、後接する形態素の情報を省略したマーカ <adv-> を用いることができる。このマーカを利用することにより、変換知識のパタン等の集約化がはかれる。同種のマーカは、<endp->, <interj->, <personpnom->, <subp-> の 4 種類がある。

*個別な情報を示すマーカ

```
(define-local-transformation /kihon-cn :j-k 3
  (
     .........
   (((:conj-form . 基本)) ((:pos . 普通名詞)))
   =>
   (1 </kihon-cn> 2)
   )
  )
```

個別な情報を示すマーカを挿入するルール。連体修飾のように、品詞情報だけで構造を作れない表現については、活用形やその他の決め手となる情報をマーカとして挿入する。このルールでは、1番めの形態素の活用形が「基本」形、2番めの形態素の品詞が「普通名詞」である場合に、間にマーカ </kihon-cn> を挿入している。これらのマーカのうち、
「基本」形を示すマーカは、
<s/kihon-cn>, <s/kihon-cn>, <auxv/kihon-cn>, <auxv/kihon-cn>, <adj/kihon-cn>, </kihon-cn>, <adj/kihon-sn>, <auxv/kihon-sn>, <auxv/kihon-sn>, </kihon-sn>, <adj/kihon-fn>, <auxv/kihon-fn>, <auxv/kihon-fn>, </kihon-fn>, <auxv/kihon-rentaishi>, </kihon-adj>, </kihon-adjnoun>, </kihon-rentaiadj>, <adj/kihon-v> の 19 種類、
「連用」形を示すマーカは、

<v/renyou-fn>, <v/renyou-sn>, <v/renyou-interj>, <adj/renyou-v>, <adj/renyou-cn>, <adj/renyou-sn>, <adj/renyou-adj>, <adj/renyou-adjnoun>, <adj/renyou-adv>, <aux/renyou-interj>, <aux/renyou-advb> の 11 種類、
その他の個別な情報を示すマーカは、
「N月1日(ツイタチ)」 表現変換のための <tsuki-hi>、本動詞に前接する並立助詞「と」を格助詞として変換するための <heijyo/to-v>、助動詞「そう(様態)」と準体助動詞「そう(伝聞)」を区別するための <kihon/sou>, <renyou/sou>, <sou/sou>、"連体助詞+体言" の <na-cn>, <na-fn>, <na-sn>, <naru-cn>, <naru-fn>, <naru-sn>、倒置・述部省略表現変換のための <adj/renyou_end>, <auxv/nichigainai_end>, <cn-kakujo/wo_end>, <fn-kakujo/wo_end>, <hantei/de_end>, <juntaiP/no_end>, <kakujo/de_end>, <npron-kakujo/wo_end>, <setsuzokuP/shi_end>, <sn-kakujo/wo_end>, <toiukoto-kakujo/de_end>、"サ変名詞+係助詞「は」+「必要」" 表現変換のための <sn+need> の 23 種類がある。

6. マーカ挿入の基本方針

マーカ挿入の基本方針は、内容語か機能語かによって異なる。

内容語とは、単独で意味を持つ語で、通常は意味コード辞書に意味コードが登録されている。変換知識においては、用例として扱われることが多い。一方、機能語は、単独では意味を持たない語であり、通常は意味コードがふられていない。以下に内容語と機能語の品詞シンボルを示す。

＜内容語＞

代名詞／普通名詞／サ変名詞／サ変形容名詞／形式名詞／形容名詞／形容詞／連体形容詞／本動詞／感動詞／ローマ字／副詞／接続副詞／数詞／準数詞

上に示した内容語が連接する場合、必ず形態素間に品詞情報を示すマーカを挿入する。ただし、本動詞・形容詞が基本形または連用形の活用をする場合は、活用形情報を示すマーカの挿入処理を先行する。

＜機能語＞

接頭辞／接続詞／連体詞／副助詞／接尾辞／人称接尾辞／助動詞／準体助動詞／係助詞／格助詞／連体助詞／接続助詞／並立助詞／準体助詞／終助詞／判定詞／補助動詞

上に示した機能語の場合、品詞の組合せや条件によってマーカを挿入するかどうかが決まる。以下に、品詞情報を示すマーカを入れる場合の組み合わせと個別なマーカを入れる場合の例を示す。

- 副助詞／終助詞／接尾辞／人称接尾辞＋内容語
- 副助詞／終助詞／接尾辞／人称接尾辞＋接頭辞／接続詞／連体詞
- 内容語＋接頭辞／接続詞／連体詞
- 「N月1日(ツイタチ)」 表現変換の際
- 本動詞に前接する並立助詞「と」を格助詞として変換する際
- 助動詞「そう(様態)」／準体助動詞「そう(伝聞)」
- 連体助詞「な」＋体言
- 倒置・述部省略表現変換の際
- "サ変名詞＋係助詞「は」＋「必要」" 表現変換の際
- 基本形の活用をする助動詞／補助動詞／準体助動詞
- 連用形の活用をする助動詞／補助動詞／準体助動詞

7. 優先順位(priority)別処理

倒置・述部省略表現変換のためのマーカを挿入するルールは1、その他は3である。優先順位をつけたい場合は、より高いあるいは低い優先順位を指定することができる。

1.4.4 total-transformation

現在、日韓翻訳では使用していない。

1.5 変換知識

1.5.1 概要

1. 目的：この翻訳システムの中心モジュールである変換処理を行なうために参照される。解析処理の済んだ入力文は、変換パタンの用例との意味距離を構成要素ごとに計算され、最小距離のパタンを用いて目的言語への構造変換が行なわれる。変換の際には、生成処理用の情報が付加され、出力データは生成処理へと渡される。

2. 管理者：解析変換知識作成者

3. 格納場所： /usr/local/TDMT/tdmt-multi-dev/transfer/j-k/p-data/*.lisp

4. 書式：

```
(define-pattern RULE-NAME :j-k CATEGORY
  (SOURCE-PATTERN) (:head N)[(CATEGORY-CONTROL)]
  =>
  (((TARGET-PATTERN) ([GENERATION-INFO]))
   (
    (EXAMPLES-LIST)
    )
   [LOCAL-DIC-LIST]
   )
  )
```

   - define-pattern ... 変換パタンを定義するための宣言。他に、define-regular-patternがある。
   - RULE-NAME ... 任意のルール名。ただし、変換知識内でユニークとする。
   - :j-k ... 翻訳の方向。この場合は日韓翻訳を示す。
   - CATEGORY ... カテゴリ(原言語構造の分類名)
   - SOURCE-PATTERN ... 原言語パタン(記述要素：変項・マーカ・日本語形態素)
   - :head N ... ヘッドの指定 (記述要素：数値)
   - CATEGORY-CONTROL ... 下部構造の制限(記述要素：変項・カテゴリ名・日本語品詞・構造分類用の集合体名)。省略可
   - TARGET-PATTERN ... 目的言語パタン(記述要素：変項・韓国語形態素)
   - GENERATION-INFO ... 生成情報(記述要素：韓国語品詞と数値)。省略可
   - EXAMPLES-LIST ... 用例(記述要素：日本語形態素)
   - LOCAL-DIC-LIST ... ローカル辞書(記述要素：変項・日韓品詞・日韓形態素)。省略可

＊詳細は、以下で説明する。

1.5.2 原言語パタン / 目的言語パタン / 用例

変換知識は、原言語表現から目的言語表現への変換をパタン形式で記述する。下の変換パタンの例では、(?x "に" ?y) が原言語パタン("に"が固定項、?x と?y が変項) であり、(!x "에" !y) や (!x "를" !y)... は目的言語パタンである。原言語パタンと目的言語パタンの対応関係は、必ずしも 1 対 1 ではなく、原言語パタン 1 つに対して、複数の目的言語パタンが考えられることが多い。そのパタンを選択する際の鍵となるのが用例である。入力文は、日本語意味コード辞書 (「1.3.1 日本語意味コード辞書」参照) を用いて用例との類似度が計算され、意味的に最も近い用例を持つ目的言語パタンが対訳として選ばれる。

```
(define-pattern kakujo-ni-np :j-k np
  (?x "に" ?y) (:head 3) (:x group-n)
  =>
  (((!x "에" !y) (padvb 2))
   (
    (("どちら") ("置く"))
    (("ホテル") ("滞在する"))
    )
   )
  (((!x "를" !y) (pobjt 2))
   (
    (("バス") ("乗る"))
    (("道") ("沿う"))
   )
  )
 )
```

例えば、ここで、「旅館に泊まる」という入力文があったとする。変項?x を「旅館」、?y を「泊まる」として、各用例との意味距離を計算する。その結果、入力文「旅館に泊まる」は (("ホテル") ("滞在する")) の用例と距離が近いということが導き出されたとすると、目的言語パタン(!x "에" !y) が選ばれる。(!x "에" !y) は、原言語の ?x の部分と ?y の部分を入れ替え、固定項の"に"を"에"に変換することを意味する。「旅館」「泊まる」という形態素単位の変換は、日韓変換辞書(「1.3.2 日韓変換辞書」参照) から対訳を得て行なわれる。(!x "에" !y) の横に記述されている (padvb 2) は生成情報(「1.5.8 生成情報」参照) で、目的言語パタンの二番目の要素が韓国語品詞、副詞格助詞であることを表している。

このような具体的な内容を伴う用例に対して、いかなる入力文にもマッチする空用例(("")) がある。以下の例のように、原言語パタンと目的言語パタンが 1 対 1 で対応するなど、構造変換に大きな影響が出ないと思われる場合は、この空用例を用いることができる。ただし、空用例と具体的な用例では、具体的な用例のほうが優先され、正確な意味距離計算が行なえるため、正しい対訳を導き出すことができる。例にある 2 ヘッドおよび 3 ヘッド用の空用例については、1.5.6「ヘッド」で説明する。

```
(define-regular-pattern dewa-reg :j-k is
  ("では" ?x) (:head 2)
  =>
  (("그럼" !x) (mconj 1))
   (
    (""))
```

```
    (("" ""))          ← 2 ヘッド用の空用例
    (("" "" ""))       ← 3 ヘッド用の空用例
   )
  )
)
```

1.5.3 パタンの種類

変換パタンの定義には2種類あり、原言語側の形態素やマーカをすべて表層形のままでマッチングする場合は define-pattern、すべて正規形でマッチングする場合は define-regular-patternを用いる。以下の例のように、define-regular-pattern で定義すると、正規形が「ていただく」である活用形(「ていただか」「ていただき」..) のすべてに対応できる。また、解析知識で正規形を「ていただく」に変更している形態素があれば、それについても同様の処理が行なえる。

```
(define-regular-pattern teitadaku-2h-reg :j-k pm
  (?x "ていただく") (:head 1 2)  ←「ていただか」「ていただき」.. とマッチングする
  =>
  (((!x "어 주" "시") (vaux1verb 2)(vaux2pend 3))
   (
    (("教える"))
    (("作る"))
   )
  )
 )
```

また、マーカについても正規形"<>"が用意されており、前後の形態素の種類に関わらず、共通に用いることができる。

```
(define-regular-pattern auxv/kihon-teiru-x-reg :j-k s+n
  (?x "ている" <> ?y) (:head 4)(:x group-p)(:y group-n)  ←マーカの正規形 < >
  =>
  (((!x "ㄴ" !y) (vaux2tend 2))
   (
    (("売る") ("店"))          "店"    = 普通名詞
    (("売る") ("ところ"))  "ところ"  = 形式名詞
   )
  )
 )
```

1.5.4 カテゴリ

各パタンの定義で指定するカテゴリは、原言語構文解析を行なう際の構造分類名であり、以下の 17 種類がある。

is(接続詞・感動詞・呼びかけ 等、文全体を修飾する語のパタン)
ss(重文のパタン)
se(文末の接続助詞・終助詞のパタン)
sm(文を受けるモダリティ表現／文末表現 のパタン)
np(体言・副詞・副詞句を格要素とする格関係／一部の接続詞 のパタン)
ap(副詞句のパタン)
sp(用言を格要素とする格関係のパタン)
pm(用言を受けるモダリティ表現／連体助詞 のパタン)
nm(体言を受けるモダリティ表現／連体助詞 のパタン)
s+m(連体修飾用言を受けるモダリティ表現・副助詞のパタン)
s+n(連体修飾用言を受ける体言・助詞のパタン)
n+n(並立助詞／名詞連続／名詞句 のパタン)
dn(接頭辞／連体詞／名詞を修飾する副詞句／派生形 のパタン)
nd(接尾辞／副助詞をはじめ変項が一つの助詞／名詞に後接する語 のパタン)
n(複合名詞のパタン)
p(用言の反復表現／合成用言 のパタン)
terminal(数詞・接頭辞を伴う頻出表現のパタン)

注) カテゴリterminalのパタンは、解析処理において:compoundの指定をして合成された語句のみに適用される。既に一つの形態素として合成されているため、構造を作る際には形態素としての扱いを受ける。

<原言語構文解析の例>

以下のように原言語構文解析が行なわれ、適用されたパタンの木構造が得られる。

```
"シングル一部屋十九万ウォンになります"
TOP [(?X ます) --- SM]
 |--?X [(?X になる) --- NM]
     |--?X [(?X <CN-CN> ?Y) --- N+N]
         |--?X [(?X <CN-CN> ?Y) --- N+N]
         |    |--?X [(シングル)]
         |    |
         |    |--?Y [(1 部屋) --- compound]
         |
         |--?Y [(190000 ウォン) --- compound]
```

"オンドルのあるホテルはどれくらいありますか"

```
TOP [(?X ます か) --- SM]
 |--?X [(?X は ?Y) --- NP]
      |--?X [(?X </KIHON-CN> ?Y) --- S+N]
      |    |--?X [(?X の ?Y) --- NP]
      |    |    |--?X [(オンドル)]
      |    |    |
      |    |    |--?Y [(ある)]
      |    |
      |    |--?Y [(ホテル)]
      |
      |--?Y [(?X <PRON-V> ?Y) --- NP]
           |--?X [(どれくらい)]
           |
           |--?Y [(あり)]
```

"電話番号は零三四四四三一七零零です"

```
TOP [(?X です) --- SM]
 |--?X [(?X は ?Y) --- NP]
      |--?X [(電話番号)]
      |
      |--?Y [(0344431700) --- compound]
```

＊1.5.9 にて「カテゴリ別変換パタン定義の基本方針とパタン例」を述べる

1.5.5 下部構造の制限

原言語構造を決定する際、ありえない構造を排除して構造の曖昧性の数を抑制するため、パタンごとに下部構造を制限することができる。下部構造の制限は、カテゴリや個々の形態素の品詞名で指定するが、それらを新たにグループ化した集合体 (以下の 7 種類) を用いると簡潔に記述できる。

category-all(全形態素および全構造)
word-all(全形態素)
group-n(体言性の形態素および構造)
word-n(体言性の形態素)
group-p(用言性の形態素および構造)
word-p(用言性の形態素)
group-s(文として認知可能な形態素および構造)

この集合体のメンバは /usr/local/TDMT/tdmt-multi-dev/transfer/j-k/dic/category-jk.lispにて定義されている。

<書式>

```
(define-category-set GROUP-NAME :j-k
  MEMBER)
```

- define-category-set ... 集合体を定義する。
- GROUP-NAME ... 集合体の名前
- :j-k ... 翻訳の方向を示す。この場合は日韓翻訳
- MEMBER ... 集合体のメンバ (記述要素 : カテゴリ・品詞シンボル)

<例>

```
(define-category-set group-n :j-k
  s+n n+n dn nd n
  普通名詞 固有名詞 代名詞 サ変名詞 形容名詞 サ変形容名詞 形式名詞 数詞 準数詞 ローマ字
  副詞 接続副詞)
```

各カテゴリが取り得る下部構造は、同じく category-jk.lispに以下のように定義されている。

<書式>

```
(define-category CATEGORY-NAME :j-k
  MEMBER)
```

- define-category ... カテゴリの定義を宣言する。
- CATEGORY-NAME ... カテゴリの名前
- :j-k ... 翻訳の方向を示す。この場合は日韓翻訳
- MEMBER ... カテゴリのメンバ (記述要素 : カテゴリ・品詞シンボル・集合体の名前)

<例>

```
(define-category np :j-k
  np sp ap pm nm s+m p word-p group-n 副詞 接続副詞)
```

例に上げたカテゴリnpは、下部構造としてnp, sp, ap, ... の構造の他、word-p, group-nで指定された構造や語句、さらに副詞・接続副詞を取り得る。変換パタンでは、これに対して制限を加え、構造の曖昧性を排除する。以下のパタンでは、(:x group-n) の指定を入れることにより、変項x が取り得る構造や語句を group-n のメンバに制限している。

```
(define-pattern kakujo-wo-np :j-k np
  (?x "を" ?y) (:head 3) (:x group-n)
  =>
  (((!x "를" !y) (pobjt 2))
   (
    (("住所") ("記入"))
    (("契約書") ("用意する"))
    )
   )
  )
```

1.5.6 ヘッド

原言語構造を決定する際、各構造において中心となる構成要素をヘッドと言い、その構造を上部の構造へ伝える重要な働きをする。指定するヘッドは、変項であっても固定項であっても構わないが、機能語単独ではヘッドになり得ない。ただし、対訳を決定するのに必要であれば、機能語が内容語を伴ってヘッドとなることがある。以下のパタンでは、(:head 1 2) を指定することにより、変項?x の内容語と機能語「ていただく」の 2 つがヘッドとして上部構造に伝搬される。

```
(define-regular-pattern teitadaku-2h-reg :j-k pm
  (?x "ていただく") (:head 1 2)
  =>
  (((!x "시") (vaux1verb 2)(vaux2pend 3))
   (
    (("乗る"))
     .........
   )
  )
 )
```

ここで、「リムジンバスに乗っていただきます」という入力文があり、以下のように原言語構造解析が行なわれたとする。

```
TOP [(?X ます) --- SM]
 |--?X [(?X に ?Y) --- NP]
     |--?X [(リムジンバス)]
     |
     |--?Y [(?X ていただく) --- PM]
         |--?X [(乗る)]
```

この文の場合は、(?x "ていただく") の上部構造(?x "ます") において、機能語「ていただく」の情報が必要になるため、2 ヘッドの指定を行っている。以下のパタンのように、「乗ります (탑니다)」の場合「ます」は現在上称形であるが、「乗っていただきます (타세요)」の場合は、「ます」を命令略待上称形として訳出しなくてはいけないからである。

```
(define-pattern masu :j-k sm
  (?x "ます") (:head 1)(:x sp np ap pm nm p s+n n+n dn nd n  word-p word-n 副詞 接続
副詞)
  =>
   (((!x "ㅂ니다") (vaux2send 2))
   (
    (("乗る"))
     .........
    )
   )
  (((!x "어요") (vaux2send 2))
   (
    (("乗る" "ていただく"))
     .........
```

```
   )
  )
 )
```

このように、(?x "ていただく") で指定された2ヘッド用例は、すぐ上部の構造 (?x "に" ?y) に伝搬されたあと、さらにそのまた上部の (?x "ます") へ、2ヘッドのまま伝搬されるため、以下のように記述する。また、空用例(「1.5.2 原言語パタン／目的言語パタン／用例」参照) の場合にも、Nヘッド用のものが用意されている。

```
(define-pattern kakujo-ni-np :j-k np
  (?x "に" ?y) (:head 3) (:x group-n)
  =>
  (((!x "를" !y) (pobjt 2))
   (
    (("リムジンバス") ("乗る" "ていただく"))
     .........
   )
  )
     .........
```

上の例は、上部構造の目的言語パタンを選択させることで対訳を決定する事例であり、その際、Nヘッドの指定が必要な場合であるが、その他、上部構造に含まれる語や文との関係から内容語や機能語の対訳を決定する際にも、Nヘッドの指定を行なっている。例えば、「三大祭りの一つに数えられています」という入力文があり、以下のように原言語構造解析が行なわれたとする。

```
TOP [(?X ます) --- SM]
 |--?X [(?X に ?Y) --- NP]
     |--?X [(?X の ?Y) --- N+N]
     |    |--?X [(3 大祭り) --- compound]
     |    |
     |    |--?Y [(一つ)]
     |
     |--?Y [(?X ている) --- PM]
         |--?X [(?X れる) --- PM]
             |--?X [(数えら)]
```

日韓翻訳の場合、機能語「れる」の対訳として一般的に、可能を表す (ㄹ 수 있), 尊敬を表す (시), そして、受動を表す (되、および その他) が考えられる (変換パタン(?x "れる") の汎用度と用例集約の点から、内容語と機能語「れる」の合成処理は行なわない。)。通常、機能語の対訳は、その機能語を含む原言語パタンを有する変換パタン内で目的言語パタンとして記述し決定するが、「れる」においては、多くの場合、上部構造に含まれる語や文との関係で意味と対訳が決定される。したがって上の例文においても、「れる」 の対訳決定は (?x "れる") の変換パタン内で行なわず、上部構造に含まれる語や文を見て決定することになる。ここでは、(?x "れる") の2つ上の構造 (?x "に" ?y) に着目し、普通名詞「一つ」と 格助詞 「に」との関係から、「れる」の対訳を決定している。対訳は下記のように (?x "に" ?y) の変換パタン内でローカル辞書を用いて与えることになる。この際、機能語「れる」の情報を上部構造の (?x "に" ?y) にまで伝搬しておかなければならないため、(?x "れる") で2ヘッドの指定をおこなっている。ローカル辞書の機能上、Nヘッドで伝搬された機能語に対訳を付与することができない

(「1.5.7 ローカル辞書」参照) ので、ここでは「数える」「れる」の組合せから、「数える」に "꾸히" という対訳を与えている。

```
(define-pattern kakujo-ni-np :j-k np
  (?x "に " ?y) (:head 3) (:x group-n)
  =>
     .........

  (((!x "로 " !y) (padvb 2))
   (
    (("一つ ") ("数える " "れる "))
   )
   (
    (y
     ((本動詞 "数える ") (vverb "꾸히 ")))
    )
   )
     .........
   )
  )
 )
```

以上２つの事例のように、Ｎヘッドの指定をする部分は、対訳を決定する際に必要となる情報である。

1.5.7 ローカル辞書

「1.3.2 日韓変換辞書」で定義された対訳はデフォルト値であり、変換知識のパタン中でローカルに別の対訳を与えることができる。例えば、日韓変換辞書の中で本動詞「ひく」の対訳は"당기"と定義しているが、「ピアノをひく」や「線をひく」の場合には、それぞれ"치"、"긋"を用いたい。このような場合は、以下のようにパタンごとの訳し分けを行なう。このローカル辞書は、上部構造で記述されるものほど優先される。

<書式>
((VARIABLE ((POS REG-EXP) TARGET)))

- VARIABLE ... 変数
- POS ... 日本語品詞シンボル
- REG-EXP ... 日本語正規形
- TARGET ... 生成情報と対訳のリスト

<例>
```
(define-pattern kakujo-wo-np :j-k np
  (?x "を" ?y) (:head 3) (:x group-n)
  =>
  (((!x "를" !y) (pobjt 2))
   (
    (("契約書") ("用意する"))
    (("方々") ("乗せる"))
    (("東京") ("する"))
    (("部屋") ("する"))
    (("椅子") ("ひく"))
     .........
    )
   )
  (((!x "를" !y) (pobjt 2))
   (
    (("ピアノ") ("ひく"))
   )
   (
    (y
     ((本動詞 "ひく") (vverb "치")))
    )
   )
  (((!x "를" !y) (pobjt 2))
   (
    (("線") ("ひく"))
   )
   (
    (y
     ((本動詞 "ひく") (vverb "긋")))
```

```
    .........
  )
 )
)
```

1.5.8 生成情報

各構造ごとに目的言語パタンが選ばれ、形態素レベルでの変換処理が済んだ入力文は、生成処理に回されるが、その際、必要に応じて生成情報が付与される。生成処理では、その情報を用いて、文法的に正しく、かつ、より自然な韓国語を出力するため、活用形の生成、叙述格助詞などの挿入 等を行なう。例えば、「奈良駅から歩いて三分です"」という入力文は、変換処理の後、以下のような形態素列データとして生成処理へ渡され、最終的に"나라역에서 걸어서 삼 분입니다."という翻訳結果が得られる。

```
*** Generation ***
 ((NPROP "나라역") (PADVB "에서") (VVERB "걷") (VAUX2TEND "어서") (NNUMBCH "3") (NCOMM "분
") (VAUX2SEND "ㅂ니다"))
 =>
 나라역에서 걸어서 삼 분입니다.
```

形態素列データ中、対訳の文字列以外の韓国語品詞名、NPROP,PADVB,VVERB,VAUX2TEND, NNUMBCH(注),NCOMM,VAUX2SEND が生成情報である。これらは、以下の変換パタンと日韓変換辞書でその付加情報として記述されている。

(注)「TDMT用韓国語形態素品詞認定基準」 株式会社コングレ 1997 で、数字に対する品詞は "数詞 NNUMB" のみとなっているが、生成処理では、算用数字を韓国語漢字数と韓国語ハングル数とに訳しわけるため、漢字数生成時 = NNUMBCH、ハングル数生成時 = NNUMBHA を生成情報として変換部で記述する。

<変換パタン>

```
(define-pattern term-desu :j-k sm
  (?x "です") (:head 1)(:x sp np ap pm nm s+m p s+n n+n dn nd n  word-p word-n 副詞 接
続副詞)
  =>
  (((!x "ㅂ니다") (vaux2send 2))
   (
     .........

(define-pattern kakujo-kara-np :j-k np
  (?x "から" ?y) (:head 3) (:x group-n)
  =>
  (((!x "부터" !y) (pauxi 2))
   (
     .........

(define-pattern setujo-te-sp
  (?x "て" ?y) (:head 3) (:x group-p)(:y sp np ap pm nm s+m p group-n word-p)
  =>
  (((!x "어서" !y) (vaux2tend 2))
     .........

(define-pattern n-fun :j-k terminal
```

```
(?x "分") (:head 2)
=>
(((!x "분") (ncomm 2))
    .........
```

＜日韓変換辞書＞

```
((普通名詞 "奈良駅") (NPROP "나라역"))
((本動詞 "歩く") (VVERB "걷 - ㄷ"))
```

*韓国語生成に関しての詳細は[山本 98]を参照のこと。

1.5.9 カテゴリ別変換パタン定義の基本方針とパタン例

＊各パタン例の目的言語パタンおよび用例は、格納場所を参照のこと。

1. **カテゴリ is**(接続詞・感動詞・呼びかけ 等、文全体を修飾する語のパタン)
   パタン種類：接続詞が固定項など、変項が一つの場合 define-regular-pattern、それ以外は define-regular-pattern または define-pattern
   ヘッド：述語の項
   変項制限：通常なし。ただし、マーカが固定項の場合等制限をつけることがある。

   (define-regular-pattern sorekara-is-reg :j-k is
   ("それから" ?x) (:head 2)
   格納場所： /usr/local/TDMT/tdmt-multi-dev/transfer/j-k/p-data/beginning.lisp
   訓練文例："それから ここにサインをお願いいたします"

   (define-regular-pattern joshi-toiukotode-is-reg :j-k is
   (?x "という" "こと" "で" ?y) (:head 5)
   格納場所： /p-rentaijo-toiu.lisp
   訓練文例："じゃあ予約はツインルームにエクストラベッド ということで 三人部屋にキャンセルが出ましたらそちらへ振り替えお願いします"

   (define-regular-pattern soretodesune-is-reg :j-k is
   ("それと" "です" "ね" <> ?x) (:head 5)
   格納場所： /marker-endp-x.lisp
   訓練文例："それとですね <> 奈良の東大寺です"

   (define-pattern marker-cn-cn-is :j-k is
   (?x <cn-cn> ?y) (:head 3)(:x 普通名詞 is ap np sp s+n n+n dn nd n)
   格納場所： /marker-cn-x.lisp
   訓練文例："お客様 <cn-cn> お客様の韓国内のご連絡先をお聞かせいただけますか"

   (define-pattern marker-interj-x :j-k is
   (?x <interj-> ?y) (:head 3)(:x 感動詞 is ap np sp)
   格納場所： /marker-interj-x.lisp
   訓練文例："はい <interj-> ご用意できます"

   (define-pattern marker-personpnom-is :j-k is
   (?x <personpnom-> ?y) (:head 3)(:x is ap np sp s+n n+n dn nd n)
   格納場所： /marker-personpnom-x.lisp
   訓練文例："はいジョンフィリップス様 <personpnom-> なんでございましょう"

   (define-pattern marker-adv-x-is :j-k is
   (?x <adv-> ?y) (:head 3)(:x 副詞 is ap np sp)
   格納場所： /adverb.lisp
   訓練文例："なるほど <adv-> そのお城の写真かなんかありますか"

2. カテゴリ ss(重文のパタン)
パタン種類：define-regular-pattern または define-pattern
ヘッド：後の文の述語の項
変項制限：通常なし

(define-regular-pattern conj-masunode-ss-reg :j-k ss
(?x "ます" "ので" ?y) (:head 4)
格納場所：/conj-node.lisp
訓練文例："朝食は七時から二階のレストランでご用意さしていただいており ますので どうぞご利用くださいませ"

(define-pattern conj-ga-ss :j-k ss
(?x "が" ?y) (:head 3)
格納場所：/conj-ga.lisp
訓練文例："自炊になります が よろしいですか"

(define-pattern joshi-desutoka-ss :j-k ss
(?x "です" "とか" ?y) (:head 4)
格納場所：/p-fukujo.lisp
訓練文例："例えば雲が少なく空気がきれいな日 ですとか あるいはまた雨上がりでお天気のよい日などです"

(define-pattern marker-endp-te-desu-ne-x-ss :j-k ss
(?x "て" "です" "ね" <endp-> ?y) (:head 6)
格納場所：/marker-endp-x.lisp
訓練文例："雨降りだしてき てですね <endp-> ちょっと荷物が多いんでどっか置く所を探してるんですよ"

(define-pattern marker-endp-x :j-k ss
(?x <endp-> ?y) (:head 3)
格納場所：/marker-endp-x.lisp
訓練文例："航空便でしょうか <endp-> 船便でしょうか"

(define-pattern marker-cn-cn-ss :j-k ss
(?x <cn-cn> ?y) (:head 3)
格納場所：/marker-cn-x.lisp
訓練文例："この青いのはいかがでしょうか大きい方で五千八百円 <cn-cn> 二百円安いですよ"

(define-pattern s/kihon-cn-end :j-k ss
(?x <s/kihon-cn> ?y) (:head 3)
格納場所：/marker-@kihon-x.lisp
訓練文例："きょうから三日間の予定で予約してます <s/kihon-cn> 鈴木といいます"

3. カテゴリ se(文末の接続助詞・終助詞のパタン)
パタン種類：通常 define-pattern

ヘッド：述語の項
変項制限：通常 (:x sp np ap pm nm p s+n n+n dn nd n word-p word-n 副詞 接続副詞)

(define-pattern end-particle-ga-se :j-k se
(?x "が") (:head 1)(:x sp np ap pm nm p s+n n+n dn nd n word-p word-n 副詞 接続副詞)
格納場所： /p-syujo.lisp
訓練文例："あいにくですがナイフとフォークは用意してございません が"

(define-pattern end-particle-naa-se :j-k se
(?x "なあ") (:head 1)(:x sp np ap pm nm p s+n n+n dn nd n word-p word-n 副詞 接続副詞)
格納場所： /p-syujo.lisp
訓練文例："おかしい なあ わたしの荷物がまだ出てこないんですけど"

(define-pattern particle-to-se :j-k se
(?x "と") (:head 1)(:x sp np ap pm nm p s+n n+n dn nd n word-p word-n 副詞 接続副詞)
格納場所： /p-syujo.lisp
訓練文例："ホテルの前の道を渡って右の方に真っ直ぐ行く と"

(define-pattern end-particle-yo-se :j-k se
(?x "よ") (:head 1)
格納場所： /p-syujo.lisp
訓練文例："この部屋は静かでよろしいです よ"

4. カテゴリ sm(文を受けるモダリティ表現／文末表現 のパタン)
パタン種類： define-regular-pattern または define-pattern
ヘッド：述語の項
変項制限：通常 (:x sp np ap pm nm s+m p s+n n+n dn nd n word-p word-n 副詞 接続副詞)

(define-pattern term-desu :j-k sm
(?x "です") (:head 1)(:x sp np ap pm nm s+m p s+n n+n dn nd n word-p word-n 副詞 接続副詞)
格納場所： /sen-de.lisp
訓練文例："はい結構 です"

(define-pattern masu :j-k sm
(?x "ます") (:head 1)(:x sp np ap pm nm s+m p s+n n+n dn nd n word-p word-n 副詞 接続副詞)
格納場所： /sen-ma.lisp
訓練文例："はいお願いし ます"

(define-pattern modal-masen-sm :j-k sm
(?x "ません") (:head 1)
格納場所： /sen-negative.lisp
訓練文例："いいえ含まれており ません"

(define-pattern tekudasai-sm :j-k sm
(?x "てください") (:head 1)(:x sp np ap pm nm s+m p s+n n+n dn nd n word-p word-n 副詞 接続副詞)
格納場所：/sen-request.lisp
訓練文例："住所を教え てください"

(define-pattern kihon_soudesu-sm :j-k sm
(?x <kihon/sou> "そう" "です") (:head 1)(:x sp np ap pm nm s+m p s+n n+n dn nd n word-p word-n 副詞 接続副詞)
格納場所：/sen-supposition-like.lisp
訓練文例："空港の免税店でしか買うことはできない <kihon/sou> そうです"

(define-pattern sn-kakujo-wo_end-sm :j-k sm
(?x "を" <sn-kakujo/wo_end>) (:head 1)(:x sp np ap pm nm s+m p s+n n+n dn nd n word-p word-n 副詞 接続副詞)
格納場所：/p-kakujo-wo.lisp
訓練文例："それでは楽しいご旅行 を <sn-kakujo/wo_end>"

(define-pattern adj/renyou_end-sm :j-k sm
(?x <adj/renyou_end>) (:head 1)(:x sp np ap pm nm s+m p s+n n+n dn nd n word-p word-n 副詞 接続副詞)
格納場所：/marker-@renyou.lisp
訓練文例："と申しましてもお一人様百五十ドル程度ですから費用の方は御心配なく <"adj/renyou_end>"

5. カテゴリ np(体言・副詞・副詞句を格要素とする格関係／一部の接続詞 のパタン)
パタン種類：通常 define-pattern、接続詞のパタンは define-regular-pattern
ヘッド：通常、述語の項
変項制限：通常、体言を格要素とする場合 (:x group-n)、副詞・副詞句を格要素とする場合 (:x 副詞 ap)、格助詞 "の" のパタン (:x group-n)(:y group-p)

(define-pattern kakujo-ga-np :j-k np
(?x "が" ?y) (:head 3) (:x group-n)
格納場所：/p-kakujo-ga.lisp
訓練文例："オンドル部屋 が よろしいですか"

(define-pattern kakarijo-ha-np :j-k np
(?x "は" ?y) (:head 3) (:x group-n)
格納場所：/p-kakarijo-ha.lisp
訓練文例："部屋代 は いくらですか"

(define-pattern kakujo-no-np :j-k np
(?x "の" ?y) (:head 3) (:x group-n)(:y group-p)
格納場所：/p-kakujo-no.lisp
訓練文例："大仏 の ある東大寺に行きたいんですがどう行けばいいんでしょうか"

(define-pattern conj-nara-np :j-k np
(?x "なら" ?y) (:head 3) (:x group-n)
格納場所：/conj-nara.lisp
訓練文例："缶ジュース なら 持ち込んでも宜しいですか"

(define-pattern marker-subp-x-np :j-k np
(?x <subp-> ?y) (:head 3) (:x group-n)
格納場所：/marker-subp-x.lisp
訓練文例："二十ドルくらい <subp-> かかると思います"

(define-pattern marker-adjnoun-v-np :j-k np
(?x <adjnoun-v> ?y) (:head 3) (:x group-n)
格納場所：/marker-adjnoun-x.lisp
訓練文例："かなり <adjnoun-v> ありますよ"

(define-pattern marker-cn-cn-np :j-k np
(?x <cn-cn> ?y) (:head 3) (:x group-n)
格納場所：/marker-cn-x.lisp
訓練文例："電話番号 <cn-cn> ちょっと探しますので"

(define-pattern marker-adv-x-np :j-k np
(?x <adv-> ?y) (:head 3) (:x 副詞 ap)
格納場所：/adverb.lisp
訓練文例："まず <adv-> 重さを計らなければなりません"

(define-regular-pattern de-np-reg :j-k np
("で" ?x) (:head 2)
格納場所：/beginning.lisp
訓練文例："今片道だけ切符を手配いたしまして で 帰りの分を向こうで手配していただくということですと十二時四十分の電車に間に合うかも分かりませんね"

(define-pattern marker-endp-ha-desu-ne-x-np :j-k np
(?x "は" "です" "ね" <endp-> ?y) (:head 6)(:x group-n)
格納場所：/marker-endp-x.lisp
訓練文例："はいこちらのツアー はですね <endp-> 京都奈良そして東京をまわるツアーになっております"

(define-regular-pattern da-to-omou-np-reg :j-k np
(?x "だ" "と" "思う") (:head 4)(:x group-n)
格納場所：/sen-think.lisp
訓練文例："そうですね比較的新しいですしきれいないいホテル だと思い ますが"

6. カテゴリ ap(副詞句のパタン)
パタン種類：通常 define-pattern
ヘッド：被修飾語の項

変項制限：副詞が固定項の場合はなし。変項の場合 (:x 副詞 ap)

(define-pattern marker-adv-x-ap :j-k ap
(?x <adv-> ?y) (:head 3)(:x 副詞 ap)
格納場所： /adverb.lisp
訓練文例：”はいまったく <adv-> 初めてです”

7. カテゴリ sp(用言を格要素とする格関係のパタン)
パタン種類： define-pattern または define-regular-pattern
ヘッド：通常、係り先の述語の項
変項制限：通常 (:x group-p)(:y sp np ap pm nm s+m p group-n word-p)

(define-pattern conj-kara-sp :j-k sp
(?x ”から” ?y) (:head 3)(:x group-p)(:y sp np ap pm nm s+m p group-n word-p)
格納場所： /conj-kara.lisp
訓練文例：”なかなかいいですねじゃあこれでお願いします九月三十日から行く から 十月一日でお願いします”

(define-pattern conj-tara-sp :j-k sp
(?x ”たら” ?y) (:head 3)(:x group-p)(:y sp np ap pm nm s+m p group-n word-p)
格納場所： /conj-tara.lisp
訓練文例：”ここを押し たら いいんですね”

(define-pattern modal-you :j-k sp
(?x ”よう” ?y) (:head 3)(:x group-p)(:y sp np ap pm nm s+m p group-n word-p)
格納場所： /modal-you.lisp
訓練文例：”かしこまりましたすぐパーティーに移れる よう あらかじめ準備しておりますので御安心くださいませ”

(define-pattern rentaijo-na-sp :j-k sp
(?x ”な” ?y) (:head 3)(:x group-p)(:y sp np ap pm nm s+m p group-n word-p)
格納場所： /p-rentaijo-etc.lisp
訓練文例：”そうですね比較的新しいですしきれい な いいホテルだと思いますが”

(define-pattern marker-endp-ka-x :j-k sp
(?x ”か” <endp-> ?y) (:head 4)(:x group-p)(:y sp np ap pm nm s+m p group-n word-p)
格納場所： /marker-endp-x.lisp
訓練文例：”そうですかどこに行けば売ってる か <endp-> ご存じですか”

(define-pattern joshi-kotoha-sp :j-k sp
(?x </kihon-fn> ”こと” ”は” ?y) (:head 5)(:x group-p)(:y sp np ap pm nm s+m p group-n word-p)
格納場所： /p-kakarijo-ha.lisp
訓練文例：”ある </kihon-fn> ことは あるんですがそのタイプは今予約でいっぱいなんですが”

(define-pattern marker-subp-x-sp :j-k sp
(?x <subp-> ?y) (:head 3) (:x group-p)(:y sp np ap pm nm s+m p group-n word-p)
格納場所：/marker-subp-x.lisp
訓練文例："そうですかごめんなさいね怒ったり <subp-> して"

(define-regular-pattern to-omou-reg :j-k sp
(?x "と" "思う") (:head 3) (:x group-p)
格納場所：/sen-supposition-like.lisp
訓練文例："そこならたぶん分かる と思い ますが"

8. **カテゴリ pm**(用言を受けるモダリティ表現／連体助詞 のパタン)
パタン種類：通常 define-regular-pattern
ヘッド：述語の項。ただし、構文の決定や原言語上部構造の訳しわけが必要な場合は、述部の項と補助用言類の複数ヘッド
変項制限：連体助詞のパタンは (:x group-s)

(define-regular-pattern teiru-reg :j-k pm
(?x "ている") (:head 1)
格納場所：/modal-aspect.lisp
訓練文例："はい持っ てい ます"

(define-regular-pattern honorific-reru-reg :j-k pm
(?x "れる") (:head 1)
格納場所：/modal-honorific.lisp
訓練文例："どちらの国から来ら れ ましたか"

(define-regular-pattern temiru-2h-reg :j-k pm
(?x "てみる") (:head 1 2)
格納場所：/modal-te.lisp
訓練文例："おもしろそうですね行っ てみ ます"

(define-regular-pattern seteitadaku-2h-reg :j-k pm
(?x "せていただく") (:head 1 2)
格納場所：/sen-itadaku.lisp
訓練文例："では確認さ せていただき ます"

(define-pattern rentaijo-na-pm :j-k pm
(?x "な") (:head 1)(:x group-s)
格納場所：/p-rentaijo-etc.lisp
訓練文例："マッコリそういう名前だったよう な 気もするんですけれど一度見せてください"

9. **カテゴリ nm**(体言を受けるモダリティ表現／連体助詞 のパタン)
パタン種類：通常 define-regular-pattern
ヘッド：体言の項。ただし、構文の決定や原言語上部構造の訳しわけが必要な場合は、述部の項と補助用言類の複数ヘッド

変項制限：通常なし

(define-regular-pattern gousei_ninaru-nm-2h-reg :j-k nm
(?x "になる") (:head 1 2)
格納場所： /modal-naru.lisp
訓練文例："料金はおいくらになりますか"

(define-pattern rentaijo-na-nm :j-k nm
(?x "な") (:head 1)
格納場所： /p-rentaijo-etc.lisp
訓練文例："有名なデパートが一軒あります"

(define-regular-pattern da-nm-reg :j-k nm
(?x "だ") (:head 1)
格納場所： /sen-de.lisp
訓練文例："どのような状況だったのですか"

10. **カテゴリ s+m**(連体修飾用言を受けるモダリティ表現・副助詞のパタン)
パタン種類： define-pattern または define-regular-pattern
ヘッド：連体修飾用言の項。ただし、構文の決定や原言語上部構造の訳しわけが必要な場合は、述部の項と補助用言類・副助詞の複数ヘッド
変項制限：通常なし

(define-regular-pattern kotonisuru-reg :j-k s+m
(?x "ことにする") (:head 1)
格納場所： /modal-suru.lisp
訓練文例："分かりましたそれじゃあタクシーで行くことにしますどうもいろいろとありがとう"

(define-regular-pattern kotoninaru-2h-reg :j-k s+m
(?x "ことになる") (:head 1 2)
格納場所： /modal-naru.lisp
訓練文例："あした泊まることになってる鈴木直子といいますが"

(define-pattern subp-tari-s+m :j-k s+m
(?x "たり") (:head 1)
格納場所： /p-fukujo.lisp
訓練文例："そちらでは江戸時代の町並みやまた実際にお客様が衣装をつけたりして日本の時代劇を直接体験していただけます"

(define-pattern subp-dake-s+m :j-k s+m
(?x "だけ") (:head 1)
格納場所： /p-fukujo.lisp
訓練文例："ちょっと見せてください電池がきれているだけかもしれませんね"

11. カテゴリ s+n(連体修飾用言を受ける体言・助詞のパタン)
パタン種類：define-pattern または define-regular-pattern
ヘッド：通常は体言
変項制限：通常 (:x group-p)(:y group-n)

(define-pattern /kihon-cn :j-k s+n
(?x </kihon-cn> ?y) (:head 3) (:x group-p)(:y group-n)
格納場所：/marker-@kihon-x.lisp
訓練文例："それでしたらゲイロードをお勧めしますインド人にも定評のある </kihon-cn> レストランなんです"

(define-pattern marker-adjnoun-cn-s+n :j-k s+n
(?x <adjnoun-cn> ?y) (:head 3) (:x group-p)(:y group-n)
格納場所：/marker-adjnoun-x.lisp
訓練文例："同じ <adjnoun-cn> お部屋ご用意できます"

(define-pattern v/renyou-fn :j-k s+n
(?x <v/renyou-fn> ?y) (:head 3) (:x group-p)(:y group-n)
格納場所：/marker-@renyou.lisp
訓練文例："見つかり <v/renyou-fn> 次第連絡します"

(define-pattern juntaijo-no-s+n :j-k s+n
(?x "の") (:head 1)
格納場所：/p-juntaijo.lisp
訓練文例："この青い の はいかがでしょうか大きい方で五千八百円二百円安いですよ"

(define-pattern rentaijo-no-s+n :j-k s+n
(?x "の" ?y) (:head 3) (:x group-p)(:y group-n)
格納場所：/p-rentaijo-no.lisp
訓練文例："はいレストランがお決まりになりましたらお出かけ の 際にはぜひフロントにお立ち寄りください周辺の地図をお渡しいたします"

(define-regular-pattern rentaijo-toiu-s+n-reg :j-k s+n
(?x "という" ?y) (:head 3) (:x group-p)(:y group-n)
格納場所：/p-rentaijo-toiu.lisp
訓練文例："それからそのゲイロードに長くいたコックさんが開いたっていうラジャ っていう レストランもあります"

(define-regular-pattern auxv/kihon-ta-hazu-reg :j-k s+n
(?x "た" <> "はず") (:head 4)(:x group-p)
格納場所：/marker-@kihon-x.lisp
訓練文例："さんがつ三日チェックイン七日チェックアウトで予約を入れて た <> はず なんですが"

(define-pattern na-cn-s+n :j-k s+n

(?x <na-cn> ?y) (:head 3)(:x group-p)(:y group-n)
格納場所：/marker-rentaijo-x.lisp
訓練文例："はい名前のとおりですねお寺の全体に金箔を貼りめぐらしておりましてとてもきらきらと光って綺麗な <na-cn> お寺でございます"

12. カテゴリ n+n(並立助詞／名詞連続／名詞句 のパタン)
パタン種類：define-pattern または define-regular-pattern
ヘッド：係り先の項(例外あり)
変項制限：通常なし

(define-pattern heijo-to-n+n :j-k n+n
(?x "と" ?y) (:head 3)
格納場所：/p-heijo.lisp
訓練文例："名前 と 住所 と 電話番号ですね日本のですか"

(define-pattern rentaijo-no-n+n :j-k n+n
(?x "の" ?y) (:head 3)
格納場所：/p-rentaijo-no.lisp
訓練文例："日本 の 連絡先でよろしいでしょうか"

(define-pattern conj-de-n+n :j-k n+n
(?x "で" ?y) (:head 3)
格納場所：/conj-de.lisp
訓練文例："奇数は東側になるのでこっちのエリア で 三つ目の角を曲がったあたりだと思いますよ"

(define-pattern kakujo-ga-n+n :j-k n+n
(?x "が" ?y) (:head 3)
格納場所：/p-kakujo-ga.lisp
訓練文例："ええ日本でも赤 が お湯青 が 水と決まってますからね"

(define-regular-pattern rentaijo-niyoru-n+n-reg :j-k n+n
(?x "による" ?y) (:head 3)
格納場所：/p-rentaijo-etc.lisp
訓練文例："船 による 入国手続きを教えてください"

(define-pattern mitaina-n+n :j-k n+n
(?x "みたいな" ?y) (:head 3)
格納場所：/sen-supposition-like.lisp
訓練文例："ナイトツアー みたいな ものありますか"

(define-pattern conj-janakute-n+n :j-k n+n
(?x "じゃ" "なく" "て" ?y) (:head 5)
格納場所：/conj-te.lisp
訓練文例："わたしトンデグ じゃなくて テグに行きたいんですけど"

(define-regular-pattern rentaijo-nikansuru-n+n-reg :j-k n+n
(?x "に関する" ?y) (:head 3)
格納場所：/conj.lisp
訓練文例："申し訳ありませんが自分の個人的な経験からどれかをお勧めするということはできませんしここには詳細な情報はホテル に関する ものしかないのです"

(define-pattern joshi-kara-made-z-n+n :j-k n+n
(?x "から" ?y "まで" ?z) (:head 5)
格納場所：/p-kakujo-kara-made.lisp
訓練文例："九月二日 から 四日 まで 三日間です"

(define-pattern rentaijo-no-koto-n+n :j-k n+n
(?x "の" "こと") (:head 3)
格納場所：/p-rentaijo-no.lisp
訓練文例："仁寺洞 のこと でしょうね"

(define-regular-pattern aruiha-n+n-reg :j-k n+n
(?x <> "あるいは" ?y) (:head 4)
格納場所：/conj.lisp
訓練文例："これは食間 <> あるいは 食前に一回四錠飲んで頂きます"

(define-pattern marker-cn-cn-n+n :j-k n+n
(?x <cn-cn> ?y) (:head 3)
格納場所：/marker-cn-x.lisp
訓練文例："はい定期観光バス <cn-cn> 予約センターにお電話なさってください"

(define-pattern marker-subp-x-n+n :j-k n+n
(?x <subp-> ?y) (:head 3)
格納場所：/marker-subp-x.lisp
訓練文例："ここからでしたら飛行機か <subp-> 船で帰ることができますよ"

13. カテゴリ dn(接頭辞／連体詞／名詞を修飾する副詞句／派生形 のパタン)
パタン種類：define-pattern または define-regular-pattern
ヘッド：係り先の項(例外あり)
変項制限：派生形のパタンは(:x 形容詞)

(define-pattern settou-yaku :j-k dn
("約" ?x) (:head 2)
格納場所：/fix-pre.lisp
訓練文例："それからツインルームは一泊十八万ウォン日本円で 約 二万五千円です"

(define-regular-pattern rentaishi-sono-reg :j-k dn
("その" ?x) (:head 2)
格納場所：/adnominal.lisp

訓練文例：”その ホテルの近くには何がありますか”

(define-regular-pattern tada-dn-reg :j-k dn
(”ただ” ?x) (:head 2)
格納場所：/beginning.lisp
訓練文例：”ただ ラビートの場合は乗り換えをしていただかないと行けませんのでちょっと難かしいと思いますが”

(define-pattern marker-adv-x-dn :j-k dn
(?x <adv-> ?y) (:head 3)
格納場所：/adverb.lisp
訓練文例：”ありますよちょうど <adv-> すぐ <adv-> 近くに有名なサンゲタンの店があります”

(define-pattern setsubi-sa :j-k dn
(?x ”さ”) (:head 1) (:x 形容詞)
格納場所：/fix-suf.lisp
訓練文例：”いいえ広 さ やグレードによって料金が変わってまいります”

14. **カテゴリ nd**(接尾辞／副助詞をはじめ変項が一つの助詞／名詞に後接する語 のパタン)
パタン種類：define-pattern または define-regular-pattern
ヘッド：内容語または係り先の項(例外あり)
変項制限：通常なし

(define-pattern setsubi-sama :j-k nd
(?x ”様”) (:head 1)
格納場所：/fix-suf.lisp
訓練文例：”ソウル大学のイヨンヒ 様 でいらっしゃいますね”

(define-pattern subp-kurai-nd :j-k nd
(?x ”くらい”) (:head 1)
格納場所：/p-fukujo.lisp
訓練文例：”二十ドル くらい かかると思います”

(define-pattern kakujo-made-nd :j-k nd
(?x ”まで”) (:head 1)
格納場所：/p-kakujo-made.lisp
訓練文例：”そこ まで はどうやって行ったらいいんですか”

(define-pattern noyou-nd :j-k nd
(?x ”の” ”よう”) (:head 1)
格納場所：/sen-supposition-like.lisp
訓練文例：”お話中 のよう です”

(define-pattern rentaijo-no-hou_kana-nd :j-k nd

(?x "の" "ほう") (:head 1)
格納場所：/p-rentaijo-no.lisp
訓練文例："はいご予定 のほう はいつでしょうか"

(define-regular-pattern rentaijo-no-hou_onsei-nd-reg :j-k nd
(?x "の" "方<ホウ>") (:head 1)
格納場所：/p-rentaijo-no.lisp
訓練文例："ホテルの前の道を渡って右 の方 に真っ直ぐ行くと"

(define-pattern marker-cn-adv_tyoudo-nd :j-k nd
(?x <cn-adv> "ちょうど") (:head 1)
格納場所：/marker-cn-x.lisp
訓練文例："それから十八時 <cn-adv> ちょうど に出発するセマウル号ならトンデグ駅に二十一時二分に到着します"

(define-pattern term-hantei/de_end :j-k nd
(?x "で" <hantei/de_end>) (:head 1)
格納場所：/sen-de.lisp
訓練文例："はいソウル で <hantei/de_end>"

15. カテゴリ n(複合名詞のパタン)
パタン種類：define-pattern または define-regular-pattern
ヘッド：係り先の項(例外あり)
変項制限：特別に付ける場合がある。

(define-pattern marker-cn-cn-n :j-k n
(?x <cn-cn> ?y) (:head 3)
格納場所：/marker-cn-x.lisp
訓練文例："ソウル大学のイ <cn-cn> ヨンヒ様でいらっしゃいますね"

(define-pattern marker-cn-sn-n :j-k n
(?x <cn-sn> ?y) (:head 3)(:x group-n)
格納場所：/marker-cn-x.lisp
訓練文例："メトロポリタン美術館はこの通りの何 <cn-sn> ブロック目ですか"

(define-pattern marker-cn-cn-n-2h :j-k n
(?x <cn-cn> ?y) (:head 1 3)
格納場所：/marker-cn-x.lisp
訓練文例："信号を三つ通りすぎたら四つ <cn-cn> めを右に曲がって下さい"
*原言語上部構造の訳しわけのため2ヘッドとする。

(define-pattern marker-cn-cn_tsuki-n :j-k n
(?x <cn-cn> "付") (:head 3)
格納場所：/marker-cn-x.lisp
訓練文例："オンドル <cn-cn> 付きの部屋がいいのですが"

(define-regular-pattern junsu-su-reg :j-k n
("数" ?x) (:head 2)
格納場所：/junsushi.lisp
訓練文例："数 百年の伝統を持つ京都の町中が古くから大切にしている行事なんですよ"

(define-pattern rentaijo-no-n :j-k n
(?x "の" ?y)(:head 3)
格納場所：/p-rentaijo-no.lisp
訓練文例："昼 の 部と夜 の 部がございますけれどもどちらの方がよろしいでしょうか"

16. カテゴリ p(用言の反復表現／合成用言 のパタン)
パタン種類：define-pattern または define-regular-pattern
ヘッド：述語の項 (例外あり)
変項制限：通常なし

(define-pattern adj/renyou-v-p :j-k p
(?x <adj/renyou-v> ?y) (:head 3)
格納場所：/marker-@renyou.lisp
訓練文例："人それぞれですが最近は飛行機を利用される方が多く <adj/renyou-v> なってきているようです"

(define-regular-pattern sugiru-2h-reg :j-k p
(?x "すぎる") (:head 1 2)
格納場所：/modal-sugiru.lisp
訓練文例："かかとが細 すぎて 安定が悪いです"

(define-pattern setujo-te-p :j-k p
(?x "て" ?y) (:head 3)
格納場所：/conj-te.lisp
訓練文例："ですけど本当にうるさく て うるさくてたまらないんですよ"

(define-regular-pattern subp-x_tari-y_tari-2h-p-reg :j-k p
(?x "たり" <> ?y "たり" "だ") (:head 1 4)
格納場所：/p-fukujo.lisp
訓練文例："今週はねずうっと雨が降っ たり <> やん だりだったんですけれども予報によるとだんだん持ち直すみたいですよ"

17. カテゴリ terminal(数詞・接頭辞を伴う頻出表現のパタン)
*解析で terminal rule 用の形態素合成を行なうこと
パタン種類：通常 define-pattern
ヘッド：通常、数量表現の場合固定項、接頭辞を伴う頻出表現のパタンの場合変項
変項制限：数詞連続パタンの場合 (:x 数詞)

(define-pattern n-ko :j-k terminal

(?x "個") (:head 2)
格納場所： /terminal-word.lisp
訓練文例："別にカートン四個があります"

(define-pattern marker-tsuki-hi :j-k terminal
(?x "月" <tsuki-hi> ?y "日")
格納場所： /terminal-word.lisp
訓練文例："八月<tsuki-hi>一日と二日の二日間です"

(define-pattern n-n-n :j-k terminal
(?x <num-num> ?y) (:head 1) (:x 数詞)
格納場所： /terminal-word.lisp
訓練文例："それから電話番号は零三の三二四五二二三三です"

(define-pattern settou-o :j-k terminal
("お" ?x) (:head 2)
格納場所： /terminal-word.lisp
訓練文例："お名前をどうぞ"

(define-regular-pattern o-v-ninaru-reg :j-k terminal
("お" ?x "になる") (:head 2)
格納場所： /terminal-word.lisp
訓練文例："百メートルほど行けばセジョン文化会館がありますすぐお分かりになると思いますよ"

1.6 各種知識の統計データ

*平成 11 年 8 月 31 日現在

辞書

1. 日本語意味コード辞書 (jma-atr-sem-code.text)
   - 形態素数 ... 18,495
   - バイト数 ... 703,450
2. 日韓変換辞書 (japanese-to-korean.lisp)
   - 形態素数 ... 11,762
   - バイト数 ... 682,335
3. 韓国語生成辞書 - 自動生成版 (korean-generation.text)
   - 形態素数 ... 1,577
   - バイト数 ... 56,788

   韓国語生成辞書 - 手作業版 (korean-generation-rev.text)
   - 形態素数 ... 37
   - バイト数 ... 1,494

解析知識

1. replace-word-info.lisp
   - 形態素数 ... 1,340
   - バイト数 ... 84,894
2. lexical-transformation
   - ルール数 ... 574
   - パタン数 ... 760
   - バイト数 ...179,420

   別途：形態素定義式 define-morph
   - ルール数 ... 28
   - パタン数 ... 28
   - バイト数 ... 3,498

	項目	ルール数	パタン数
	形態素の合成	205	216
	形態素の分割	1	1
	品詞の変更	3	3
目的別内訳	正規形の変更	85	91
	表層形の変更	166	259
	誤りの修正	2	3
	terminal rule	112	187
	計	574	760

3. local-transformation

   - ルール数 ... 142
   - パタン数 ... 153
   - バイト数 ... 23,066

4. total-transformation

   - ルール数 ... 0
   - パタン数 ... 0
   - バイト数 ... 0

変換知識

- ルール数 ... 1,299
- バイト数 ... 897,360

1. パタンの種類別内訳

パタンの種類	個数
define-pattern	765
define-regular-pattern	534
計	1,299

2. カテゴリ別内訳

カテゴリ	個数
is	142
ss	61
se	18
sm	188
np	146
ap	1
sp	112
pm	170
nm	24
s+m	11
s+n	60
n+n	86
dn	42
nd	61
n	13
p	16
terminal	148
計	1,299

第 2 章

解析変換知識作成手順

以下の章では、解析変換知識の作成手順を

- 意味距離計算を使わずに訓練する場合
- 意味距離計算を使って訓練する場合

のそれぞれの場合で例文を用いて解説する。

システムのデフォルトは、平成 11 年 8 月 31 日時点で、以下のような値になっている。

- 訓練の単位 ... 発声単位
- 意味距離計算 ... ON
- 新部分翻訳 ... ON

新部分翻訳が ON になっているため、翻訳訓練に際しては、空用例以外の exact match 用例(入力文と完全に一致する用例) をすべて追加し、望ましい翻訳結果が得られた時点で、

- 文分割(CONN) が起きていない場合は、距離がほぼ 0 になっていること
- 文分割 (CONN) が起きている場合は、分割位置の妥当性に問題がなく、距離がほぼ 5 の倍数分（文分割の数 x5）離れていること

を訓練終了の条件として確認するものとする。

作業にあたっては TDMT 翻訳知識作成ツールを利用するため、前もって、[河井 94]を参照のこと。

2.1 意味距離計算を使わずに訓練する場合

2.1.1 解析変換知識作成手順

ここでは、意味距離計算を行なわない場合の解析変換知識作成手順を説明する。作業は以下の流れで行い、既に訓練済みの知識を利用できる場合は、省略して次のステップに進む。

1. 前準備
   ⇓
2. 翻訳実行
   ⇓
3. 形態素情報の確認
   ⇓
4. lexical-transformation の作成
   ⇓
5. local-transformation の作成
   ⇓
6. 日韓変換辞書の作成
   ⇓
7. 変換パタンの新規作成 および、目的言語パタンと用例の追加
   ⇓
8. 韓国語生成辞書の作成
   ⇓
9. 訓練終了

2.1.2 例 "宿泊料についてですが消費税五百円はひかせていただきます"

1. 前準備

   (a) allegro common lisp の起動。

   (b) システムの起動ならびにルールとタギングデータのロード。
   setup-jk-uttr.lisp をロードすることにより、まとめて処理。

   (c) パッケージを翻訳モードに切替え。

   (d) 距離計算を OFF に切替え。

   (e) TDMT 翻訳知識作成ツールの起動。

2. 翻訳実行

   TDMT ツールのメニューで翻訳を実行する。
   新部分翻訳が ON になっているため、部分的に解析変換知識と用例が充足されている箇所は翻訳結果が得られるが、完全な翻訳ではない。

```
*** Input ***
 宿泊料についてですが消費税五百円はひかせていただきます
*** Output ***
 ..................................... 당겨 드리겠습니다. (50.0)
```

3. 形態素情報の確認

   TDMT ツールのメニューで、 Morph Analysis Only を選択すると以下のようにタギングデータが表示される。

```
sentence: 宿泊料についてですが消費税五百円はひかせていただきます
  WORD       REG-EXP        POS      CONJ-FORM PRON         SEM-CODES
----------+----------+----------+----------+----------+---------------
宿泊料      宿泊料      普通名詞    ---         ---         748
について    について    格助詞      ---         ---         ---
です        です        判定詞      基本        ---         ---
が          が          接続助詞    ---         ---         ---
消費税      消費税      普通名詞    ---         ---         749
500         0 0         数詞        ---         ---         120
円          円          普通名詞    ---         ---         828
は          は          係助詞      ---         ---         ---
ひか        ひく        本動詞      ない        ---         262 263 335 387 428 450 871
せ          せる        助動詞      た          ---         ---
ていただき  ていただく  助動詞      連用        ---         ---
ます        ます        助動詞      基本        ---         ---
```

   次に Morph Analysis を選択して比較し、適用されている解析知識を確認する。

```
sentence: 宿泊料についてですが消費税五百円はひかせていただきます
  WORD       REG-EXP        POS      CONJ-FORM PRON         SEM-CODES
----------+----------+----------+----------+----------+---------------
宿泊料      宿泊料      普通名詞    ---         ---         748
について    について    格助詞      ---         ---         ---
です        です        判定詞      基本        ---         ---
が          が          接続助詞    ---         ---         ---
消費税      消費税      普通名詞    ---         ---         749
500         0 0         数詞        ---         ---         120
円          円          普通名詞    ---         ---         828
は          は          係助詞      ---         ---         ---
ひか        ひく        本動詞      ない        ---         262 263 335 387 428 450 871
せていただき せていただく 助動詞      連用        ---         ---
ます        ます        助動詞      基本        ---
```

ここでは、対訳の決定上、「せていただく」が既に合成されていることがわかる。方針としては、さらに「N円」を terminal rule用に合成することにする。

4. lexical-transformationの作成

方針に従って、以下のルールを作成する。

```
(define-lexical-transformation num-yen :j-k 5
  (
   (((:pos . 数詞)) ((:word . "円")))
   =>
   (lex ((1 2) (:pos . 普通名詞) (:reg-exp . "N円") (:compound 1 2)))
   )
  )
```

作成したルールをコンパイルし、再度翻訳を実行して形態素情報を確認すると、以下のように表示される。

```
sentence: 宿泊料についてですが消費税五百円はひかせていただきます
  WORD       REG-EXP        POS      CONJ-FORM PRON         SEM-CODES
----------+----------+----------+----------+----------+---------------
宿泊料      宿泊料      普通名詞    ---         ---         748
について    について    格助詞      ---         ---         ---
です        です        判定詞      基本        ---         ---
が          が          接続助詞    ---         ---         ---
消費税      消費税      普通名詞    ---         ---         749
500円       N円         普通名詞    ---         ---         121 828
は          は          係助詞      ---         ---         ---
ひか        ひく        本動詞      ない        ---         262 263 335 387 428 450 871
せていただき せていただく 助動詞      連用        ---         ---
```

ます　　ます　　助動詞　　基本　　---　　---

5. **local-transformationの作成**

この文では、マーカ挿入の基本方針に従って、「消費税」普通名詞 <> 普通名詞「N円」の境界を示すマーカを挿入する。そのため、以下のルールを作成する。

```
(define-local-transformation cn-cn :j-k 3
  (
   (((:pos . 普通名詞)) ((:pos . 普通名詞)))
   =>
   (1 <cn-cn> 2)
   )
  )
```

作成したルールをコンパイルし、再度翻訳を実行すると <CN-CN> というマーカが挿入される。

```
LOCAL-TRANSFORMATION ...
(宿泊料 について です が 消費税 <CN-CN> 500円 は ひか せていただき ます)
```

6. **日韓変換辞書の作成**

この文で対訳が必要な語は、「宿泊料」,「消費税」,「500円」,「ひか(ひく)」。部分的に翻訳を行なうと、

```
*** Input ***
(宿泊料)
*** Output ***
숙박료. (0.0)

*** Input ***
(消費税)
*** Output ***
(0.0)

*** Input ***
(500円)
*** Output ***
(0.0)

*** Input ***
(ひか)
*** Output ***
당깁니다. (0.0)
```

← 「ㅂ니다」は韓国語生成処理で文末に自動的に付加される(詳細は [山本98] を参照のこと)。

となり、「消費税」,「500 円」の対訳が不足していることがわかるが、「500 円」は、terminal rule用に合成しているので、変換パタンで対訳をあたえなければならない。「消費税」は、日韓変換辞書 japanese-to-korean.lisp に以下のように追加する。

```
((普通名詞 "消費税") (NCOMM "소비세"))
```

また、「500 円 は ひく」なので、「ひか(ひく)」の対訳は"깍"にしたいが、これは変換パタンにローカル辞書で追加することにする。

日韓変換辞書に登録した「消費税」の部分を再度部分的に翻訳すると、以下のように対訳が得られる。

```
*** Input ***
(消費税)
*** Output ***
 소비세. (0.0)
```

7. 変換パタンの新規作成 および、目的言語パタンと用例の追加

部分的な翻訳を行ない、翻訳結果が得られない場合は翻訳メニューの Search Rule Name で変換パタンを検索する。該当する変換パタンが存在しない場合は変換パタンを新規作成し、存在する場合は、正しい翻訳結果が得られるように必要に応じて目的言語パタンを追加し、不足用例を補う。

この文の場合、先に terminal rule用に合成した「500 円」の変換パタンを以下のように作成してから、部分的な翻訳を行なう。

```
(define-pattern n-yen :j-k terminal
  (?x "円") (:head 2)
  =>
  (((!x "엔") (ncomm 2))
   (
    (("0 0"))
    (("0 0列"))
   )
  )
 )
```

「500 円」の部分を翻訳すると、以下のように対訳が得られる。

```
*** Input ***
(500 円)
*** Output ***
 오백 엔. (0.0)
```

部分的な翻訳は、各形態素の係り受けを考えながら取り得る下部構造の数が少ない構造から行なうので、この文の場合、以下の順となる。

(a) (宿泊料 について)

(b) (宿泊料 について です)

(c) (消費税 <CN-CN> 500 円)

(d) (ひか せていただき)

(e) (500 円 は ひか せていただき)

(f) (ひか せていただき ます)

(g) (宿泊料 について です が 消費税 500 円 は ひか せ ていただき ます)

```
*** Input ***
(宿泊料) (について)
*** Output ***                          ← "について" の変換パタン検索結果、無し。
 ......... (5.0)

*** Input ***
(宿泊料) (について) (です)
*** Output ***                          ← "です" の変換パタン検索結果、有り。
 ............... (10.0)

*** Input ***
(消費税) (500 円)
*** Output ***                          ← <cn-cn> の変換パタン検索結果、無し。
 ......... (5.0)

*** Input ***
(ひか) (せていただき)
*** Output ***                          ← "せていただく" の変換パタン検索結果、有り。
 당겨 드립니다. (0.0)

*** Input ***
(500 円) (は) (ひか) (せていただき)
*** Output ***                          ← "は" の変換パタン検索結果、有り。
 ..........당겨 드립니다. (10.0)

*** Input ***
(ひか) (せていただき) (ます)
*** Output ***                          ← "ます" の変換パタン検索結果、有り。
 당겨 드리겠습니다 (0.0)

*** Input ***
(宿泊料) (について) (です) (が) (消費税) (500 円) (は) (ひか) (せていただき) (ます)
*** Output ***                          ← "が" の変換パタン検索結果、有り。
 ..................................당겨 드리겠습니다. (50.0)
```

結果、(?x "について"),(?x <cn-cn> ?y) の変換パタン、および (?x "です"),(?x "は" ?y),(?x "が" ?y) の用例が不足していることがわかるので、変換パタンの新規作成、用例の追加を行なう。不足している (?x "について"),(?x <cn-cn> ?y) の変換パタンを以下のようにそれぞれ作成し、用例を登録する。

```
(define-pattern joshi-nituite-nd :j-k nd  ←変換パタンの新規作成
  (?x "について") (:head 1)
  =>
  (((!x "에" "대하" "어서") (padvb 2)(vverb 3)(vaux2tend 4))
   (
    (("宿泊料"))    ←用例を登録
   )
  )
 )


(define-pattern marker-cn-cn-n+n :j-k n+n  ←変換パタンの新規作成
  (?x <cn-cn> ?y) (:head 3)
  =>
  (((!x !y))
   (
    (("消費税") ("N円"))    ←用例を登録
    )
   )
  )
```

(?x "です") の変換パタンに不足用例を補う。

```
(define-pattern term-desu :j-k sm
  (?x "です") (:head 1)(:x sp np ap pm nm s+m p s+n n+n dn nd n  word-p word-n 副詞 接続副詞)
  =>
  (((!x "ㅂ니다") (vaux2send 2))
   (
    (("宿泊料"))    ←用例の追加
     .........
    )
   )
  )
```

(?x "は" ?y) の変換パタンに不足用例を補う。このとき、ローカル辞書の登録を同時に行なう。

```
(define-pattern kakarijo-ha-np :j-k np
  (?x "は" ?y) (:head 3) (:x group-n)
  =>
```

```
(((!x "는" !y) (psubj 2))
 (
  (("こちら") ("Nドル"))
   .........
  )
 )
(((!x "는" !y) (psubj 2))
 (
  (("N円") ("ひく" "せていただく"))    ←用例(2ヘッド用例)の追加
  )
  (
   (y
    ((本動詞 "ひく") (vverb "깎")))    ←ローカル辞書の登録
    )
   )
  )
```

(?x "が" ?y) の変換パタンに不足用例を補う。

```
(define-pattern conj-ga-ss :j-k ss
  (?x "が" ?y) (:head 3)
  =>
  (((!x "만" !y) (pconj 2))
   (
    (("宿泊料") ("ひく" "せていただく"))    ←用例(2ヘッド用例)の追加
     .........
      )
     )
    )
```

不足している変換パタンを作成し かつ、すべての不足用例を補うと、以下のように全文の翻訳結果が得られる。

```
*** Input ***
 宿泊料についてですが消費税五百円はひかせていただきます
*** Output ***
 숙박료에 대하서입니다만 소비세 오백 엔은 깎아 드리겠습니다. (0.0)
```

8. 韓国語生成辞書の作成

上掲の翻訳結果では、動詞 vverb "대하" と 転成連結語尾 vaux2tend "어서" がうまく接続できていないので韓国語生成辞書で動詞 vverb "대하"を検索してみる。もし登録されていなければここで korean-generation-rev.text に以下のように登録する。登録済にもかかわらず正しく生成が行われない場合は、システム管理者に報告する。

("대하" VVERB (change 여 - 불규칙))

.....

韓国語生成辞書をロードし、再度翻訳を実行すると翻訳結果は以下のようになる。

*** Input ***
宿泊料についてですが消費税五百円はひかせていただきます
*** Output ***
숙박료에 대해서입니다만 소비세 오백 엔은 깎아 드리겠습니다. (0.0)

9. 訓練終了

TDMT ツールの画面では、上の Output を得るために、以下の解析知識、変換パタン、日韓変換辞書が適用されたことが表示される。

```
*** Distance calculation in Transfer ***
    0.000000 :: (CONJ-GA-SS) SS :(?X が ?Y) => (!X 만 !Y) .... ((宿泊料) (ひく せていただく))
    0.000000 :: (TERM-DESU) SM :(?X です) => (!X 비니다) .... ((宿泊料))
    0.000000 :: (JOSHI-NITUITE-ND) ND :(?X について) => (!X 에 대하 어서) .... ((宿泊料))
    0.000000 :: (~DIC) : (普通名詞 宿泊料) => ((NCOMM 숙박료))
    0.000000 :: (MASU) SM :(?X ます) => (!X 겠 비니다) .... ((ひく せていただく))
    0.000000 :: (KAKARIJO-HA-NP) NP :(?X は ?Y) => (!X 는 !Y) .... ((N円) (ひく せていただく))
    0.000000 :: (MARKER-CN-CN-N+N) N+N :(?X <CN-CN> ?Y) => (!X !Y) .... ((消費税) (N円))
    0.000000 :: (~DIC) : (普通名詞 消費税) => ((NCOMM 소비세))
    0.000000 :: (N-YEN) TERMINAL :(?X 円) => (!X 엔) .... ((0 0))
    0.000000 :: (~DIC) : (数詞 0 0) => ((NNUMBCH 500))
    0.000000 :: (SETEITADAKU-2H-REG) PM :(?X せていただく) => (!X 어 드리) .... ((ひく))
    0.000000 :: (~LOCAL) : (本動詞 ひく) => ((VVERB 깎))
TOTAL DISTANCE =  0.000000

*** Distance calculation in analysis ***
LEXICAL-TRANSFORMATION ...
    0.000000 :: (GOUSEI-SETEITADAKU) 3 : (せ ていただき) => (せていただき)
    0.000000 :: (NUM-YEN) 5 : (500 円) => (500円)
LOCAL-TRANSFORMATION ...
    0.000000 :: (CN-CN) 3 : (消費税 500円) => (消費税 <CN-CN> 500円)
TOTAL DISTANCE =  0.000000
```

2.2 意味距離計算を使って訓練する場合

2.2.1 解析変換知識作成手順

ここでは、意味距離計算を行なう場合の解析変換知識作成手順を説明する。作業は以下の流れで行い、それぞれのステップで確認事項に問題がある場合は、2.1「意味距離計算を使わずに訓練する場合」の手順に従って対処する。

1. 前準備
   ⇓
2. 翻訳実行
   ⇓
3. 形態素情報の確認
   ⇓
4. 構文解析の確認
   ⇓
5. 翻訳結果の確認
   ⇓
6. exact match 用例の追加
   ⇓
7. 訓練終了

2.2.2 例”ここにそれでは名前と電話番号と予約番号を書いておきますねありがとうございました”

1. 前準備

   (a) allegro common lisp の起動。

   (b) システムの起動ならびにルールとタギングデータのロード。
   setup-jk-uttr.lisp をロードすることにより、まとめて処理。

   (c) パッケージを翻訳モードに切替え。

   (d) TDMT 翻訳知識作成ツールの起動。

2. 翻訳実行

   TDMT ツールのメニューで翻訳を実行する。
   新部分翻訳が ON になっており、かつ意味距離計算を ON にしているため、出力は得られるが、望ましい翻訳ではない。

```
*** Input ***
 ここにそれでは名前と電話番号と予約番号を書いておきますねありがとうございました
*** Output ***
 여기에..그럼 이름과 전화 번호와 예약 번호를 쓰죠..감사합니다. (15.4888935)
```

   ".."は、翻訳知識が不足している箇所を示す。

3. 形態素情報の確認

   TDMT ツールの Morph Analysis を選択する。

```
sentence: ここにでは名前と電話番号と予約番号を書いておきますねありがとうございました
  WORD       REG-EXP        POS      CONJ-FORM PRON         SEM-CODES
----------+----------+----------+----------+----------+---------------
ここ         ここ         代名詞      ---         ---         101 156
に           に           格助詞      ---         ---         ---
それでは     それでは     接続詞      ---         ---         ---
名前         名前         普通名詞    ---         ---         822
と           と           並立助詞    ---         ---         ---
電話番号     電話番号     普通名詞    ---         ---         823
と           と           並立助詞    ---         ---         ---
予約         予約         サ変名詞    ---         ---         448
番号         番号         普通名詞    ---         ---         823
を           を           格助詞      ---         ---         ---
書い         書く         本動詞      た          ---         339 807
ておき       ておく       助動詞      連用        ---         ---
ます         ます         助動詞      基本        ---         ---
ね           ね           終助詞      ---         ---         ---
ありがとうございました ありがとうございました 感動詞      ---         ---         476 696
```

ここでは、

(a) タギングデータに誤りがないか
→あれば担当者に報告するが、すぐには修正できないのでパッチを当てる。

(b) 意味コードの登録に誤りや不足がないか
→あれば担当者に報告し、修正あるいは追加してもらう。

(c) replace-wordの登録に誤りや不足がないか
→あれば担当者に報告し、必要に応じて日本語を原言語とする解析変換知識作成者で協議する。

(d) lexical-transformationに誤りや不足がないか
→あれば該当知識の修正や追加を行なう。

を確認する。この文については問題ない。

4. **構文解析の確認**

原言語構文解析結果を確認し、不足している知識を検討する。

```
TOP [(?X ?Y) --- :CONN]
 |--?X [(?X ?Y) --- :CONN]
 |   |--?X [(?X に) --- ND]
 |   |   |--?X [(ここ)]
 |   |
 |   |--?Y [(それでは ?X) --- IS]
 |       |--?X [(?X ます ね) --- SM]
 |           |--?X [(?X を ?Y) --- NP]
 |               |--?X [(?X と ?Y) --- N+N]
 |               |   |--?X [(?X と ?Y) --- N+N]
 |               |   |   |--?X [(名前)]
 |               |   |   |
 |               |   |   |--?Y [(電話番号)]
 |               |   |
 |               |   |--?Y [(?X <SN-CN> ?Y) --- N+N]
 |               |       |--?X [(予約)]
 |               |       |
 |               |       |--?Y [(番号)]
 |               |
 |               |--?Y [(?X ておく) --- PM]
 |                   |--?X [(書い)]
 |
 |--?Y [(ありがとうございました)]
```

ここでは、格助詞「に」において、(?x に ?y) のパタンを選べず、「ここ」と下部構造の関係を作ることに失敗していることがわかる。理由は、カテゴリ NP の (?x "に" ?y) が取り得ないカテゴリ IS("それでは" ?x) が下部構造にあるからである。したがって、カテゴリNP の ("それでは" ?x) を新規作成することにする。

```
(define-regular-pattern soredeha-np-reg :j-k np  ←変換パタンの新規作成
  ("それでは " ?x) (:head 2)
  =>
  ((("그럼 " !x) (mconj 1))
   (
    (("")          ← 空用例
    (("" ""))      ← 2ヘッド用の空用例
    (("" "" ""))   ← 3ヘッド用の空用例
   )
  )
 )
```

再度翻訳すると、以下のような構文解析が得られる。

```
TOP [(?X ?Y) --- :CONN]
 |--?X [(?X ます ね) --- SM]
 |   |--?X [(?X に ?Y) --- NP]
 |       |--?X [(ここ)]
 |       |
 |       |--?Y [(それでは ?X) --- NP]
 |           |--?X [(?X を ?Y) --- NP]
 |               |--?X [(?X と ?Y) --- N+N]
 |               |   |--?X [(?X と ?Y) --- N+N]
 |               |   |   |--?X [(名前)]
 |               |   |   |
 |               |   |   |--?Y [(電話番号)]
 |               |   |
 |               |   |--?Y [(?X <SN-CN> ?Y) --- N+N]
 |               |       |--?X [(予約)]
 |               |       |
 |               |       |--?Y [(番号)]
 |               |
 |               |--?Y [(?X ておく) --- PM]
 |                   |--?X [(書い)]
 |
 |--?Y [(ありがとうございました)]
```

5. 翻訳結果の確認

翻訳結果は以下のようになる。

```
*** Input ***
 ここにそれでは名前と電話番号と予約番号を書いておきますねありがとうございました
*** Output ***
 여기에 그럼 이름과 전화 번호와 예약 번호를 쓰죠.. 감사합니다. (11.388894)
```

上掲の翻訳結果では、依然、翻訳知識が不足している箇所を示す ".." が入り、構文解析の当該部分には文分割の:CONNが起きているが、文分割の以下の仕様に合致しているので、分割位置は妥当であると判断し、正しい構文解析が得られたものとする。

〔文分割の妥当性〕

- 文として認知可能な構造を分断するカテゴリ、SS あるいは IS の代わりに用いられている。
- 分割された文要素が独立したもの（文や呼びかけ等）として成り立っている。
- 分割された文要素同士が主従関係になっていないと解釈できる。
- 上記の用件を満たせば、リーフにマーカが来ても正しい構造とする。

ここで、準備的動作を表す "아 / 어 놓" , 意志約束 (中称) を表す "～ ㄹ게요" を生成したいので (?x "ておく") に "書く" の用例を、(?x "ます" "ね") に ((!x "ㄹ게" "요") (vaux2send 2) (pauxi 3)) の目的言語パタンと "書く" "ておく" の用例をそれぞれ追加する。

```
(?x "ておく")

(define-regular-pattern teoku-2h-reg :j-k pm
  (?x "ておく") (:head 1 2)
  =>
  (((!x "어 놓") (vaux1verb 2))
   (
    (("キープする"))
    (("送る"))
    (("予約する"))
    (("書く"))     ←用例の追加
   )
  )
  (((!x))
   (
    (("調べる"))
   )
  )
 )

- - - - - - - - - - - - -

(?x "ます" "ね")

(define-pattern masune :j-k sm
  (?x "ます" "ね") (:head 1)(:x sp np ap pm nm s+m p s+n n+n dn nd n  word-p word-n 副詞 接続副詞)
  =>
  (((!x "죠") (vaux2send 2))
```

```
  (
   (("数える " "れる "))
   .........
  )
 )
 (((!x "겠 " "네 " "요 ") (vaux2pend 2)(vaux2send 3)(pauxi 4))
  (
   (("ピーシークラス " "になる "))
  )
 )
 (((!x "ㄹ게 " "요 ") (vaux2send 2) (pauxi 3))  ←目的言語パタンの追加
  (
   (("書く " "ておく "))                          ←用例の追加
  )
 )
)
```

以下のような翻訳結果が得られるので、これを最終結果とする。

```
*** Input ***
 ここにそれでは名前と電話番号と予約番号を書いておきますねありがとうございました
*** Output ***
 여기에 그럼 이름과 전화 번호와 예약 번호를 써 놓을게요..감사합니다. (7.288894)
```

ただし、(?x "ます" "ね") で追加した目的言語パタン((!x "ㄹ게 " "요") (vaux2send 2) (pauxi 3)) は、この訓練文については問題ないが、自由入力文では文構造によって不適切な翻訳になる可能性もある。このような語尾類の決定等の精度を上げるためには、文脈処理や文中の情報を用いた処理等を利用する必要がある。

6. exact match 用例の追加

翻訳結果が決まったので、以下の適用パタン一覧を見て exact match 用例の不足を補う。

```
*** Distance calculation in Transfer ***
    5.000000 :: (CONN) CONN :(?X ?Y) => (!X !Y) .... NIL
    0.000000 :: (MASUNE) SM :(?X ます ね) => (!X ㄹ게 요) .... ((書く ておく))
    0.900000 :: (KAKUJO-NI-NP) NP :(?X に ?Y) => (!X 에 !Y) .... ((横) (差し込む ておく))
    0.000000 :: (~DIC) : (代名詞 ここ) => ((NPRON 여기))
    0.000005 :: (SOREDEHA-NP-REG) NP :(それでは ?X) => (그럼 !X) .... (( ))
    1.055556 :: (KAKUJO-WO-NP) NP :(?X を ?Y) => (!X 를 !Y) .... ((時間) (調べる ておく))
    0.333333 :: (HEIJO-TO-N+N) N+N :(?X と ?Y) => (!X 와 !Y) .... ((電話番号) (名前))
    0.000000 :: (HEIJO-TO-N+N) N+N :(?X と ?Y) => (!X 와 !Y) .... ((名前) (電話番号))
```

```
    0.000000 :: (~DIC) : (普通名詞 名前) => ((NACTV 이름))
    0.000000 :: (~DIC) : (普通名詞 電話番号) => ((NCOMM 전화 번호))
    0.000000 :: (MARKER-SN-CN-N+N) N+N :(?X <SN-CN> ?Y) => (!X !Y) .... ((予約) (番
号))
    0.000000 :: (~DIC) : (サ変名詞 予約) => ((NACTV 예약))
    0.000000 :: (~DIC) : (普通名詞 番号) => ((NCOMM 번호))
    0.000000 :: (TEOKU-2H-REG) PM :(?X ておく) => (!X 어 놓) .... ((書く))
    0.000000 :: (~DIC) : (本動詞 書く) => ((VVERB 쓰))
    0.000000 :: (~DIC) : (感動詞 ありがとうございました) => ((INTER 감사합니다))
TOTAL DISTANCE = 7.288894
```

用例が不足しているのは、文分割の (CONN) と空用例以外で距離が"0.000000"でない箇所である。(?X に ?Y),(?X を ?Y),(?X と ?Y) のパタンにそれぞれ用例を追加する。

7. 訓練終了

念のため、意味距離計算を OFF にして、翻訳結果を出力する。同様の結果が出なければ、どこかのパタンに exact match 用例が不足していることになるので、追加する。

参考文献

[河井 94] 河井 淳：ＴＤＭＴ翻訳知識作成ツールの利用方法 TR-IT-0088 (1994)

[古瀬 95] 古瀬蔵・増井淳子：言語データベースの概要 TR-IT-0136 (1995)

[溝口 98] 溝口淳子：ＴＤＭＴ用日本語形態素仕様明細 (1998)

[山本 98] 山本和英：ＴＤＭＴ韓国語生成部 仕様書 (1998)

索引