TR-IT-0328

TDMT's German Generation Module

Michael Paul

2000.02

概要

TDMT is a multi-lingual approach for translating spoken language dialogues in order to allow free communication between people who use different languages. We are using a transfer-driven machine translation approach which integrates example-based and rule-based translation in a common framework. This report will give an overview of the TDMT system. Its focus, however, is the German generation module.
The empirical linguistic knowledge provided by the application of the transfer rules to the morphological analysis of the Japanese input sentence forms the input of the generation process. Given the transfer result each linguistic constituent will be analyzed both on the word level (inflection phenomena) and on the sentence level (selection of a topological field).
On the word level we make use of a classification-based word analysis approach, which is used to generate well-formed surface words in the generation output. In order to handle the rather free word order of the German language in an accurate way, we use a *topological field* approach, whereby a sentence model consists of several "fields", whose linear composition specifies the chosen word order within the sentence.

ATR Interpreting Telecommunications Research Laboratories

Contents

1 Introduction 1
1.1 Cooperative Integrated Translation 1
1.2 System Architecture 2
1.3 Translation Example 10

2 German Peculiarities 14
2.1 Inflectional Phenomena 14
2.2 Free Word Order 15
2.3 Clause Types 15

3 German Word Model 17
3.1 Classification 18
3.1.1 Lexems 18
3.1.2 Suffixes 23
3.2 Encoding 24
3.3 Segmentation 26
3.4 Inflection 27

4 German Sentence Model 29

5 Generation Module 30
5.1 Terminology 31
5.2 Generation Dictionary 32
5.3 Preprocess 34
5.4 Analysis 35
5.5 Inflection 42
5.6 Postprocess 43
5.7 Interface to Synthesis 43

6 Evaluation 45
6.1 The Evaluation Procedure 46
6.2 Results 46
6.3 Progress of JG 47
6.4 Pending Problems 47

References 50

Appendix 52

A Morphology 52

B Transfer 53

C German Word Analysis/Inflection **59**
C.1 Lexem Definition 59
C.2 Suffix Tree 60
C.3 Top-level Functions 61
C.4 Inflection Functions 62
C.5 Visualization Functions 66

D Generation **68**

List of Figures

1 TDMT's System Architecture 2
2 Source Structure Candidates 12
3 Best Target Structure 13
4 Topological Fields of the German Sentence Structure 15
5 Suffix Tree "N" 24
6 IAL Tree for Suffix "EN" 24
7 Clarification Dialog 25
8 Segmentation Algorithm 27
9 Generation Algorithm 28
10 Creation of the Generation Dictionary 33

List of Tables

1 Unknown Words 6
2 Noun Classification 19
3 Irregular Verb Stem Classification 19
4 Subclass Classification of Irregular Verbs 20
5 Verb Classification 21
6 Adjective Classification 22
7 Ordinal Classification 23
8 Topological Fields in TDMT 29
9 Syntax of the Transfer Result 33
10 Size of the Generation Dictionary 35
11 Syntax of the Generation Input 36
12 Analysis of Unknown Words 37
13 Correspondence between Topological Fields and Top-Level Constituents 42
14 Transfer Knowledge Statistics 46
15 Quality and Time 46
16 Transfer Knowledge Statistics for JG 47
17 Quality Improvement of JG 47
18 Japanese Part-of-Speeches 52
19 Bigram Markers 53
20 Rule Categories 53
21 Rule Patterns 54
22 German Part-of-Speeches 68
23 German Part-of-Sentences 68
24 Generation Markers 70
25 German Sentence Types 70
26 Inflection Attributes 71
27 Part-of-Speech Inflection 72

1 Introduction

Within the context of current research activities at ITL "Towards Cross-Language Global Communications" we are concerned with the translation of naturally spoken utterances in order to allow free communication between people who use different languages.

A practicable machine translation system for dialogue translations has to fulfill two main requirements. First, it has to carry out a quick translation, because long pauses would disturb the flow of the conversation. Second, the system has to be able to handle the characteristics of spoken language, because humans tend to speak rather ungrammatical during a conversation. Thus ill-formed utterances has to be handled in an appropriate way.

In this chapter we will describe the principle ideas of our multi-lingual *chat translation* system. Then we give a short overview of the single modules and a translation example in order to illustrate the flow of information required to achieve an appropriate translation of spoken language utterances.

1.1 Cooperative Integrated Translation

The transfer-driven machine-translation (TDMT) approach, developed at ATR, uses a *constituent boundary parsing* method in an exampled-based framework, which simultaneously executes structural parsing and the application of transfer knowledge by means of pattern matching, in order to handle the incremental translation of multi-lingual spoken language input (cf. [Furuse & Iida 96], [Sumita et al. 99]).

A *pattern* expresses a meaningful unit of the linguistic structure of the input sentence and is defined as a sequence consisting of variables ("placeholder" for sentence constituents) separated by symbols representing the constituent boundaries. These boundaries are *functional words*, i.e. frequently occurring surface words, which modify or relate contents words, e.g. the Japanese particle を.

However, due to the spoken language input these kind of grammatical entities are often omitted in a conversation. Therefore we have to recover this ungrammaticality in order to be able to translate ill-formed utterances as well. In TDMT this is done by using *part-of-speech bigram* markers (cf. Appendix B, Table 19), which will be inserted in a preprocessing step into the input when no surface word divides an expression into its constituents, e.g. in the case of compound nouns a <*cn-cn*> marker will be inserted between the successive common nouns in order to match the phrase as a compound.

In order to limit the explosion of structural ambiguity during parsing, the patterns are attached to several linguistic levels (sentence, noun phrase, terminals, etc.), whereby the instantiations of variables are restricted to the same or lower level.

The incremental pattern matching algorithm is based on the idea of *chart parsing* and carried out in a bottom-up and left-to-right fashion. According to the category hierarchy all possible pattern rules of this level are matched against the input. For each match the semantic distance between the examples in our training database and the current variable bindings of this rule are calculated in order to find the closest match for these patterns in the given context. The iteration of these matching algorithm gives us all possible segmentations and semantic distances of the respective input subparts. In order to restrict the search space for the best sentence parse, we are using only the best substructures (minimal semantic distance) for the further segmentation process. Finally the minimization of the overall semantic distance for each succeeded parse determines the best sentence structure based on the knowledge in our example database.

Simultaneous with the structural parsing TDMT applies its transfer knowledge to the instantiations of the segmented input parts. The transfer knowledge associates each pattern of the source language with a corresponding linguistic expression of the target language, based on the empirical knowledge obtained during the processing of the training data. The target expressions consist of the word translations of the instantiated pattern variables enriched with linguistic constituent expressions of the target language.

Thus, by choosing the most appropriate substructures, the transfer component delivers the target expressions most consistent with TDMT's empirical knowledge to the generation module.

1.2 System Architecture

TDMT consist of three main components: the *morphological analysis* for each of the source languages, the central *transfer* component and a *generation* module for all target languages as listed in Figure 1.

Given the respective knowledge sources we can handle the bilingual translation between Japanese/English (JE,EJ) and Japanese/Korean (JK,KJ) and the translation from Japanese to German (JG) and from Japanese to Chinese (JC) by using a unique approach.

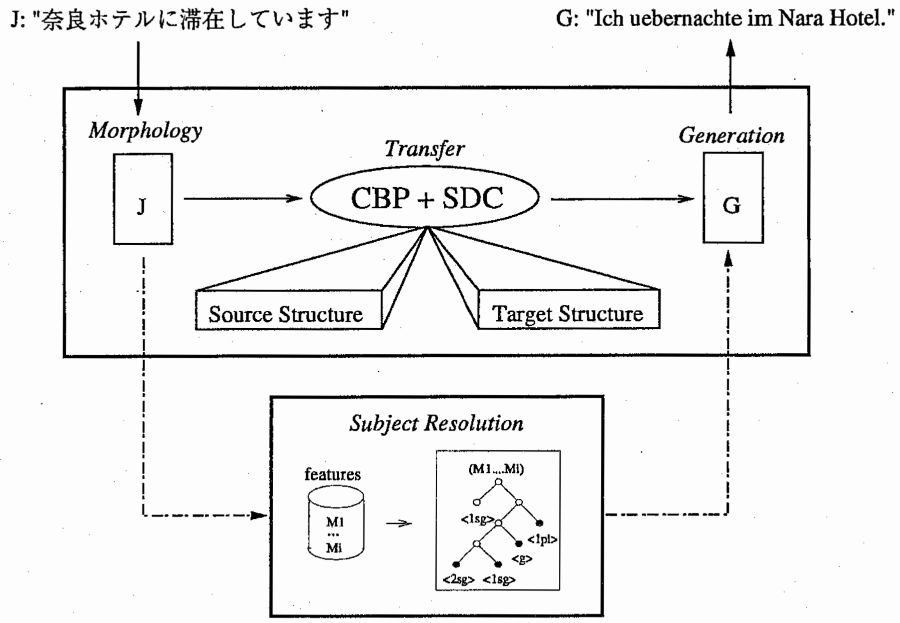


Figure 1: TDMT's System Architecture

Input Data

The input languages of the current translation system are Japanese, English and Korean. The input is a simple string representing a single utterance and either consists of the ITL speech recognizer output (cf. [Singer 97]) or is typed in using a keyboard.

Morphology

The morphological analysis component is only used for free keyboard input. The analysis of the input string is based on *n-gram* models trained on the ATR speech database (cf. [Furuse & Masui 95]). Each word in the morphological dictionary is matched against the input resulting in a set of possible segmentations of the specified sentence. According to the *n-gram* models the most probable segmentation is chosen. The result of the morphological module is a list of morphological components consisting of the surface morpheme, a part-of-speech tag, the semantic category of the word and some additional attributes.

Transfer

The knowledge sources used in the constituent boundary parsing (CBP, [Furuse & Iida 96]) approach are a source-to-target word dictionary, a set of pattern rules, which are (so far) created by hand based on the analysis of the given training sentences, and a thesaurus component for the source language, which is used for the semantic distance calculation (SDC, [Sumita & Iida 92]).

These knowledge sources have to be compiled for each source-target language pair. Depending on how much time was spent working on the different translation pairs, the size of the dictionaries and the amount and contents of the training sets[1] differ between the translation modes. The current numbers are listed in section 6,Table 14.

Transfer Dictionary

The dictionary used for each language pair consists of 1-to-1 translation entries for the words occurring in the ATR speech database. The format of the transfer dictionary is defined as:

```
(define-transfer-dic mode t
    {( (spos sstr) {( tpos tstr )}+ )}*
    ...
    )
```

{}*	zero or more elements
{}+	one or more elements
mode	translation mode {:j-g :j-e :e-j :j-k :k-j}
spos	source language part-of-speech
sstr	source language word (a string)
tpos	target language expression
tstr	target language word (a string)

Below you find an example of the contents of the JG dictionary. The specification of the target language expressions are described in more detail in Chapter 5 and Appendix D.

```
(define-transfer-dic :j-g t
    ((普通名詞 "一般") (ADJEKTIV "generell"))
    ((普通名詞 "アフタヌーンツアー") (NOMEN "Nachmittagstour"))
    ((普通名詞 "のぞみ") (EIGENNAME "Nozomi" :gender MAS))
    ((普通名詞 "打合わせ") (VERB "besprechen"))
    ((本動詞 "応急処置する")
      (AKK-OBJ {OHNE} (ADJEKTIV "erste")(NOMEN "Hilfe"))(VERB "leisten"))
    ...)
```

[1] The transfer component for JG is described in more detail in [Paul & Yamazaki 99]

In two preprocessing steps the morphological analysis results are modified in order to align the transfer input data with the syntax specified in the transfer dictionary and pattern rule set. Besides this redefinition of morphological compounds, additional structure information based on the analyzed part-of-speech tags is inserted into the input. This enables us not only to handle complex sentences, but also to recover ill-formed utterances in an appropriate way.

Lexical Transformation

The lexical transformation of the transfer input tries to map the morphologically analyzed utterance segments to the syntax which is required by the transfer knowledge sources (transfer dictionary and rule set).

```
(define-lexical-transformation lexical-rule-name mode priority
   {(
    ( {( {( skey . sval )}+ )}+ )
    ⇒
    (
     {(lex ( ({num}+) {( tkey . tval )}+ ) }*
     | {( { @morph-rule-name }* {num}* )}*
    )
   )}+
   ))

(define-morph @morph-rule-name mode ({num}* {str}*)
   {( tkey . tval )}+
   )

(define-string $string-rule-name mode ({num}* {str}*)
   {( :word . num )}+
   )
```

{}*	zero or more elements
{}+	one or more elements
"X | Y"	either X or Y
lexical-rule-name	name of the lexical transformation rule (a symbol)
mode	translation mode { :j-g :j-e :e-j :j-k :k-j }
priority	priority level of the rule (a number)
(*skey* . *sval*)	= (:word . *str*) (:reg-exp . *str*) (:pos . *part-of-speech*)
(*tkey* . *tval*)	= (:word . *str*) (:reg-exp . *str*) (:pos . *part-of-speech*) (:all-copy . *num*) (:sem-code (*sem-code*)) (:compound . {*num*}+)
str	= "..."
num	number, which refers to the n-th component of the source-expression
sem-code	thesaurus category (*str* or *num*)
part-of-speech	(cf. Appendix A, Table 18)

The lexical transformation rules are used on the one hand to correct errors of the morphological analysis and on the other hand to create new morphological compounds, which can be matched by the defined rules. For example in the JG rule `num-jikan` below, a number plus the noun "時間" (*Zeit,time*) is combined to the new common noun "N時間" with its respective part-of-speech (:pos) and semantic category (:sem-code) information. Thus given an appropriate transfer rule, the input "四時間" should be translated as *"vier Stunden"* (*four hours*), instead of *"vier Zeiten"* (*four times*).

```
(define-lexical-transformation num-jikan :j-g 6
    ((((:reg-exp . " 0 0 "))((:word . "時間")))
    ⇒
    ((lex ((1 2) (:pos . 普通名詞) (:reg-exp . "N時間") (:sem-code ("150"))
              (:compound 1 2))))
    ))
```

The *define-morph* rules are used during the lexical transformation preprocess as macros to modify the morphological analysis. For example in TDMT the part-of-speech サ変名詞 classifies a noun, which can be used in combination with the verb "する" as a full verb. However, in the transfer dictionary we can assign only one translation to each source entry (part-of-speech plus word expression in its regular form). Thus if we want to disambiguate the nominal and verbal use of a サ変名詞 like "予約" (*reservation*, *to reserve*) we have to apply a lexical transformation rule in order to assign the correct part-of-speech.

```
(define-morph @add-suru :j-g (1 "する")
  (:all-copy . 1)(:pos . 本動詞)
  )

(define-lexical-transformation sahen-verb-wo-suru :j-g 3
  ((((:pos . サ変名詞)) ((:word . "を")) ((:reg-exp . "する")))
   ⇒
   ((@add-suru))
  ))
```

Using the rules listed above the phrase "予約をする", which has to be translated in the verbal context, is transformed to the full verb "予約する", which forms a separate entry in the dictionary:

```
((サ変名詞 "予約") (NOMEN "Reservierung"))    ;; reservation
((本動詞 "予約する") (VERB "reservieren"))     ;; to reserve
```

The *define-string* rules are used to modify the morphological analysis result by combining the respective strings, e.g. for handling nominalizations of adjectives. In the example below the adjective "大きい" followed by the suffix "さ" will be combined to the noun "大きさ", which can be added to the transfer dictionary as a regular entry.

```
(define-string $word-1 :j-g (1)
    (:word . 1)
    )

(define-lexical-transformation gousei-adj-sa :j-g 2
    ((((:pos . 形容詞))((:pos . 接尾辞)(:word . "さ")))
    ⇒
    (lex ((1 2)(:pos . 普通名詞) (:reg-exp $word-1 2) ))
    ))
```

Unknwon Word Transformation

If words occuring in the input are not in the dictionary, they are marked as unknown as listed in Table 1. We distinguish three different types. If the source word contains only digits in combination with special characters like hyphens or brackets the expression is treated as a numerical expression, e.g. the phone number "(0774)95-1308". If the unknown word starts with one or more digits and ends in a sequence of letters we assume a unit expression, e.g. "100cc". Otherwise, the expression is handled as a proper noun, e.g. "AT&T".

Table 1: Unknown Words

source "..."	source marker	semantic code
digits only	数詞 " ^ アラビア"	("120")
digits + letters	普通名詞 " ^ アラビア記号"	("121" "828")
letters only	普通名詞 " ^ 頭字語	("999" "316" "713")

Local Transformation

The local transformation rules are used to insert additional markers into the morphological analysis result in order to parse compound phrases, e.g. sequences of initials, as one unit and to handle ill-formed utterances in an appropriate way.

Whenever the analyzed segments are not separated by functional words, we have to insert additional constituent boundaries between the input segments in order to be able to analyze the correct structure.

```
(define-local-transformation local-rule-name mode priority
   {(
    ( {( {( skey . sval )}+ )}+ )
    ⇒
    (
     (({num}+ marker {num}+ )
      ( {example }* )
      ))
   )}+
   )
```

{}*	zero or more elements
{}+	one or more elements
local-rule-name	name of the local transformation rule (a symbol)
mode	translation mode { :j-g :j-e :e-j :j-k :k-j }
priority	priority level of the rule (a number)
(*skey* . *sval*)	= (:word . *str*) (:reg-exp . *str*) (:pos . *part-of-speech*) (:conj-form . *part-of-speech*)
str	= "..."
num	number, which refers to the n-th component of the source-expression
marker	bigram marker (cf. Appendix B, Table 19)
example	list of example expressions for semantic distance calculation
part-of-speech	(cf. Appendix A, Table 18)

According to the analyzed part-of-speech tags of the successive segments, a bigram marker, i.e. the concatenation of the part-of-speech tags with a "–" as delimiter encapsulated in "< ... >" brackets, will be inserted between the segments. The set of bigram markers used in the current version of JG is listed in Appendix B, Table 19.

For example, the utterance ”タクシーに皮ジャケット置き忘れました” contains two particle omissions: the particle ”の” of the phrase ”皮のジャケット” and the particle ”を” of the phrase ”ジャケットを置き忘れる”. Given the grammatical correct sentence ”タクシーに皮のジャケットを置き忘れました” we can match the patterns ”の” and ”を” as constituent boundaries of the input sentence. However, in the spoken language utterance we have to rely on the part-of-speech tags in the input sequence for the definition of the appropriate local transformation rules:

```
(define-local-transformation cn-cn :j-g 4
  ((((:pos . 普通名詞)) ((:pos . 普通名詞)))
   ⇒
   (1 <CN-CN> 2)
   ))
```

```
(define-local-transformation cn-v :j-g 3
  ((((:pos . 普通名詞)) ((:pos . 本動詞)))
   ⇒
   (1 <CN-V> 2)
   ))
```

Applying these rules to the given input we can recover the original sentence structure using the <CN-CN> and <CN-V> bigram markers as follows:

```
          タクシー   に      皮       ジャケット  置き忘れ   まし     た
          普通名詞  格助詞  普通名詞  普通名詞   本動詞   助動詞  助動詞
                                      ⇓
タクシー   に      皮      <CN-CN>  ジャケット  <CN-V>  置き忘れ   まし     た
普通名詞  格助詞  普通名詞   (の)    普通名詞    (を)    本動詞   助動詞  助動詞
```

Transfer Rules

After the preprocessing of the morphological analysis result described above, we have to analyze the structure of the input sentence by matching the enriched segmentation of the input sentence against all patterns defined in our rule set (cf. Appendix B, Table 21). These pattern rules are defined as follows:

```
(define-pattern pattern-name mode cat
  (pattern) (:head num) ( vkey vcat )*
  ⇒
  {( ( tpattern {( texp )}* )
     ( { example }* )
     ( {( var
          ( (spos sstr ) { texpl }+ )
          )}*
     ) )}+ ... )
```

```
(define-regular-pattern pattern-name-reg mode cat
  (pattern) (:head num) ( vkey vcat )*
  ⇒
  {( ( tpattern {( texp )}* )
     ( { example }* )
     ( {( var
          ( (spos sstr ) { texpl }+ )
          )}*
     ) )}+ ... )
```

{}*	zero or more elements
{}+	one or more elements
mode	translation mode { :j-g :j-e :e-j :j-k :k-j }
cat	category level at which a rule is defined
pattern	sequence of *var*'s and constituent boundaries
str	= ”. . .”
num	number, which refers to the n-th component of the source-expression (in the list before ”⇒”)
marker	bigram marker (cf. Appendix B, Table 19)
var	name of a variable representing some unknown part of the input
vkey	use var name as a keyword, e.g. :x for VAR x
vcat	category restriction for *var* binding
example	list of example expressions for semantic distance calculation
tpattern	translation of *pattern* (sequence of *var*'s and constituent boundaries)
texp	list of target language expressions (numbers refer to elements in *tpattern*)
texpl	list of target language expressions
spos	part-of-speech of Japanese word (cf. Appendix A, Table 18)
sstr	source language word (a string)

The header of each pattern rule consists of a rule name (*pattern-name*), the translation mode of this rule (*mode*), the rule category (*cat*) and the pattern itself. The category determines the linguistic level (cf. Appendix B, Table 20), at which the rule can be applied. The additional *:head* option determines which variable instantiations of this pattern are used for the semantic distance calculation with the examples in our database, whereby the type of entities which can be bound to the *vkey* variable is restricted by the value of *vcat*.

The rule body specifies all possible translation contexts for the given pattern based on the analysis of the training data, i.e. the body of a pattern rule consists of several translation rules, which specifies the chosen target expressions (translation of the pattern) in the context of the training sentences together with the respective examples, whereby the number and order of the examples agree with the number and order of the pattern variables.

Additionally, local dictionary entries can be specified in the body of each translation rule. For the analyzed input part *(spos sstr)*, which is bound to *var*, the target expressions specified in { *texpl* }+ will be used for the translation, thus overwriting the respective transfer dictionary entry.

In the case of pattern rules of the type *define-pattern* the constituent boundaries are matched exact. However, in the case of *define-regular-pattern* rules not only the word of the boundary, but also its regular form is matched against the input, which is used as well for matching inflection suffixes, e.g. "ている" against "ていれ ば", as for matching kanji and hiragana circumscriptions, e.g. "かもしれません" against "かも知れません".

If a pattern can be matched against an input part, the semantic distance between the variable instantiations and all examples specified for this pattern rule are calculated. In the case of an *exact match* the distance is 0. However, instead of an example an empty string can be specified in the body of the transfer rule. These *default rules* are used whenever the list of examples would be too large. The semantic distance is slightly higher than 0 in order to prefer exact matches.

The context of the best match (minimal semantic distance) is chosen for the translation of the matched input part, i.e. the rule of the closest example is used for the translation of this source pattern.

Given the following two sentences of the JG training sentences we want to illustrate the selection of the best translation context based on our training data.

> "十月十日にワシントンの方 に 友達と行きたいんですけど"
> (I'd like to go to Washington on October the tenth with a friend)
> ⇒ *Ich möchte gerne am zehnten Oktober mit meinem Freund nach Washington fahren*
>
> "十分おきにトイレ に 行きます"
> (I'm going to the toilet every ten minutes)
> ⇒ *Ich gehe alle zehn Minuten auf die Toilette*

In both sentences the functional word "に" is used as a constituent boundary, whereby the pattern *(?x "に" ?y)* is defined as below.

```
(define-pattern kakujo-ni-np :j-g np
    (?x "に" ?y) (:head 3)
    ⇒
    (((!y "nach" !x)(PP (PRAEPOSITION 2) 3 :case DAT))
        ( (("ワシントン") ("行く")) ))
    (((!y "auf" !x)(PLACE (PRAEPOSITION 2) 3 :case AKK))
        ( (("トイレ") ("行く")) )))
```

In the context of ”ワシントンの方に行く” the pattern will be translated as a prepositional phrase using the preposition ”nach”. However in the context of ”トイレに行く” the preposition ”auf” is chosen. Given this transfer knowledge the unknown sentence ”東京に行きます” (*I'll go to Tokyo*) would be matched against the first example, because the semantic distance between (”東京” ”行く”) and (”ワシントン” ”行く”) is closer than the distance between (”東京” ”行く”) and (”トイレ” ”行く”). Thus the preposition ”nach” (instead of ”auf”) would be used for the translation *“Ich fahre nach Tokyo”*.

Incremental Pattern Application

The incremental parsing algorithm [Furuse & Iida 96] is based on a chart parsing method using a bottom-up and left-to-right strategy. The parsing is carried out by combining active and passive arcs, whereby a *passive* arc is created, if all variables of a pattern are instantiated or a fully instantiated sub-pattern can be matched against the input parts. An *active* arc is created if the left part of the pattern can be matched, but the right variables are still not instantiated.

Whenever a passive arc satisfies the leftmost part of an uninstantiated variable in the pattern of an active arc, the patterns are combined to a substructure by instantiating the variable with the passive arc. In order to restrain the number of competing structures only the best substructures (minimal semantic distance) are used for the combination of arcs if a new passive arc has to be created. This procedure is iterated until a new active arc cannot be created.

The semantic distance of the complete source structure is computed by summing up the distances of the patterns multiplied with some weights [Sumita & Iida 92].

Simultaneously to the parsing we build up the corresponding target tree by combining the selected target expressions, whereby the leaves of the target tree consists of the word translation of the source tree leaves according to the transfer dictionary.

The parsing result of an input utterance can be ambiguous. Thus we have to select the best parse by minimizing the overall semantic distance of each source structure candidate in order to find the most suitable context based on the knowledge of our training data.

Subject Resolution

The ellipsis resolution module incorporated into the TDMT system resolves omitted subjects for the translation of Japanese utterances into English and German (cf. [Paul & Yamamoto 00]).

For the resolution task we utilize a decision-tree approach. To learn the referential relations from a training corpus we have chosen a C4.5[2]-like machine learning algorithm without pruning.

The input of the resolution module consists of all features of the morphological analyzed utterance. By parsing down the decision tree according to these attributes the resolution result is obtained as the ellipsis tag contained in the parse leaf (cf. table 24). The omitted subject is than recovered in the generation module by adding a appropriate target expression to the target structure.

Generation

The output of the transfer module consists of the empirical linguistic knowledge of the selected target tree, whereby the tree structure is represented in an encapsulated list format.

[2] cf. [Quinlan 93]

In the generation module, which will be described in detail in Chapter 5, this list structure will be recursively analyzed and the linguistic information is used to generate the translation output.

Output Data

The target languages in the current system are Japanese, Korean, English and German. The generated translation of the input utterance is a simple string. However, within our chat translation demonstration system we make use of the synthesis module CHATR, developed at ITL (cf. [Weeks 97]). In order to synthesize the translation in an appropriate way we have to make the best out of all knowledge sources available during the translation process. The structure of the interface is basically the same for all language, but the information differs from language to language.

For German the interface contains not only the translation string, but also the complete morphological analysis of each word and the structure of the German sentence (cf. Chapter 5.7).

1.3 Translation Example

To illustrate the flow of information during the translation process we want to give you an example of a short sentence, which is not in the training set.

Input Data

”この車を四時間借りたいのですが”

(”I’d like to rent this car for four hours”)

Morphology

Besides the word and the respective part-of-speech attributes the Japanese word model contains further information about its normalized word form (*regular form*) and its semantic category[3]. Numbers are transformed to a digit representation, but still handled as single morphemes. The morphological analysis results in the following segmentation of the input sentence.

この	車	を	4	時間	借り	たい	の	です	が
連体詞	普通名詞	格助詞	数詞	普通名詞	本動詞	助動詞	準体助詞	判定詞	接続助詞
この	車	を	0 0	時間	借りる	たい	の	です	が
–	997	–	120	152	–	–	–	–	–

Lexical Transformation

However, using the lexical transformation rule below the number plus the following noun are combined by creating a new compound morpheme with the part-of-speech ”普通名詞” (common noun) and the regular form ”N時間”. These are used for matching the new temporal phrase against given examples during the incremental pattern application.

```
(define-lexical-transformation num-jikan :j-g 6
   ((((:reg-exp . ” 0 0”)) ((:word . ”時間”)))
    ⇒
    (lex ((1 2) (:pos . 普通名詞) (:reg-exp . ”N時間”) (:compound 1 2)))
    ))
```

[3] The semantic codes are based on [Ono & Hamanishi 81]

After the modification of the morphological analysis the segmentation of the input sentence consists of the following parts:

この	車	を	4時間	借り	たい	の	です	が
連体詞	普通名詞	助詞	普通名詞	本動詞	助動詞	準体助詞	判定詞	接続助詞
この	車	を	N時間	借りる	たい	の	です	が
–	997	–	828	374	–	–	–	–

Local Transformation

Analyzing the successive part-of-speech tags of the given input sentence, we recognize, that there is no constituent boundary dividing the compound noun "4時間" and the verb "借りる". In order to analyze the correct structure of the sentence, we have to separate this constituents by adding an additional boundary to the input. According to the respective part-of-speeches a <CN-V> marker is inserted into the segmentation using the local transformation rule listed below:

```
(define-local-transformation cn-v :j-g 3
   ((((:pos . 普通名詞)) ((:pos . 本動詞)))
   ⇒
   (1 <CN-V> 2)
   ))
```

After the preprocessing steps are completed, the input of the incremental pattern application algorithm consists of the following sentence morpheme segmentation.

この	車	を	4時間	<CN-V>	借り	たい	の	です	が
連体詞	普通名詞	助詞	普通名詞		本動詞	助動詞	準体助詞	判定詞	接続助詞
この	車	を	N時間		借りる	たい	の	です	が
–	997	–	828	–	374	–	–	–	–

Transfer Rules

Each pattern of our rule set is matched against the input resulting in several possible parses. For our example we have chosen the two best matches in order to illustrate the further processing. Besides the matched patterns and the respective rule categories the main constituents of the variable instantiations as well as the respective examples of our database with its semantic distance values are listed below.

During the first parse the following patterns are matched, whereby the overall semantic distance is calculated as 0.97779584.

pattern	category	pattern head ↔ matched example	distance
(?X "が")	SE	("借りる") ↔ ((""))	(0.000005)
(?X "ん です")	SM	("借りる") ↔ (("借りる"))	(0.000000)
(?X "を" ?Y)	NP	(("車") ("借りる")) ↔ (("フィルム") ("くださる"))	(0.777778)
("この" ?X)	DN	(("車")) ↔ (("車"))	(0.000000)
(?X <CN-V> ?Y)	NP	(("N時間")("借りる")) ↔ (("N日間") ("借りる"))	(0.200008)
(?X "時間")	TERM	((" 0 0")) ↔ ((" 0 0"))	(0.000000)
(?X "たい")	PM	(("借りる")) ↔ ((""))	(0.000005)

Thereby three exact matches and two *default matches* could be achieved. However due to the quite far semantic distance of the phrase "車を借りる" and the example "フィルムをくださる" these structure seems to be not very appropriate for a good translation[4]. The second parse matching the patterns below is semanticly closer (distance=0.64445746).

pattern	category	pattern head ↔ matched example	distance
(?X "ん です が")	SM	("借りる") ↔ (("頂く"))	(0.444444)
(?X "を" ?Y)	NP	(("車") ("借りる")) ↔ (("車") ("借りる"))	(0.000000)
("この" ?X)	DN	(("車")) ↔ (("車"))	(0.000000)
(?X <CN-V> ?Y)	NP	(("N時間")("借りる")) ↔ (("N日間") ("借りる"))	(0.200008)
(?X "時間")	TERM	((" 0 0")) ↔ ((" 0 0"))	(0.000000)
(?X "たい")	PM	(("借りる")) ↔ ((""))	(0.000005)

Incremental Pattern Application

The combination of the local pattern matches are combined as described in Chapter 1.2 and the results of both possible source structures are listed in Figure 2, whereby the second one is selected as the best source structure. For this structure each pattern is translated in the context of the selected examples. The following target expressions are used for the translation of the respective patterns.

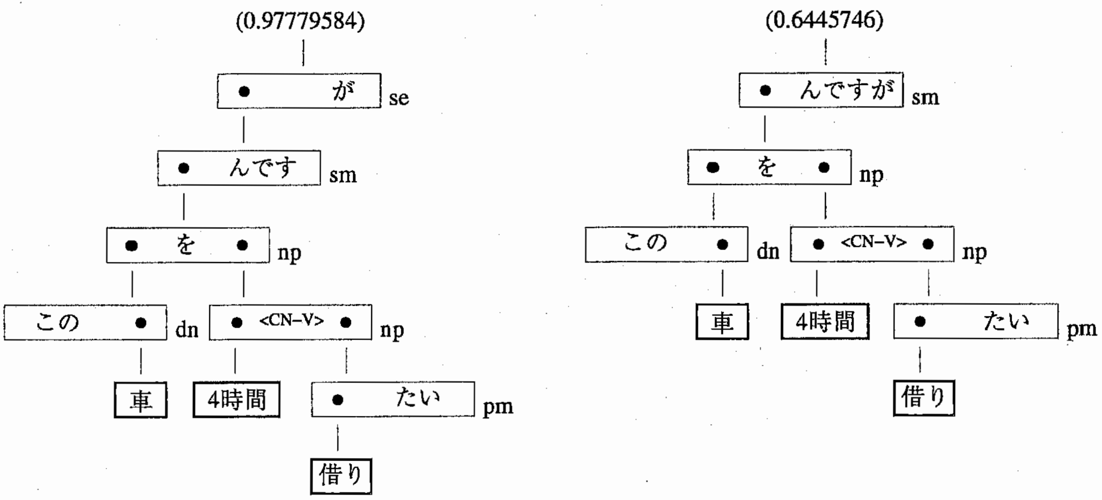


Figure 2: Source Structure Candidates

pattern	target expression
(?X "ん です が")	[!X]
(?X "を" ?Y)	[!Y (AKK-OBJ !X)]
("この" ?X)	[(NP (DETERMINATIV "dieser/diese/dieses") !X)]
(?X <CN-V> ?Y)	[(PP (PRAEPOSITION "für") !X :case AKK) !Y]
(?X "時間")	[(NP !X (NOMEN "Stunde"))]
(?X "たい")	[(VP (AP+ (ADVERB "gerne")) (MODALVERB "mögen"))]

dictionary	target expression
(普通名詞 "車")	[(NOMEN "auto")]
(数詞 " 0 0")	[(KARDINALZAHL "4")]
(本動詞 "借りる")	[(*AKK-OBJ (PERSONALPRONOMEN "er/sie/es")) (VERB "leihen")]

[4] This structure would be translated as "*Bitte mögen Sie mir gerne dieses Auto für vier Stunden leihen*" (*please could you lend the car for four hours to me*).

Simultaneous to the combination of the matched source patterns the target structure is combined resulting in the structure listed in Figure 3.

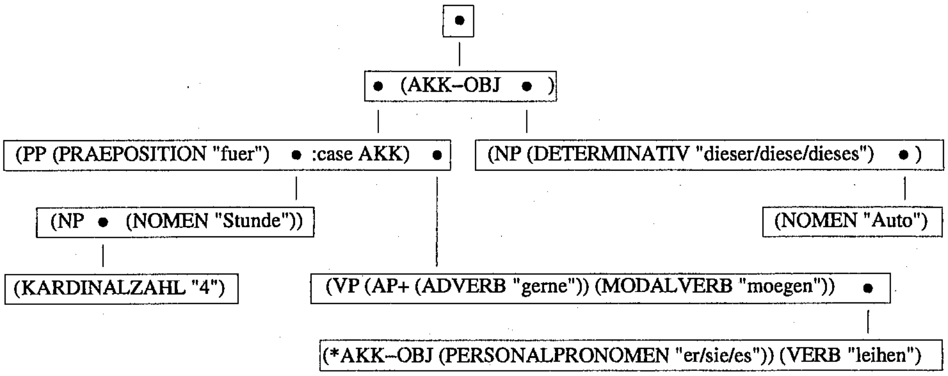

●
● (AKK-OBJ ●)
(PP (PRAEPOSITION "fuer") ● :case AKK) ●
(NP (DETERMINATIV "dieser/diese/dieses") ●)
(NP ● (NOMEN "Stunde"))
(NOMEN "Auto")
(KARDINALZAHL "4")
(VP (AP+ (ADVERB "gerne")) (MODALVERB "moegen")) ●
(*AKK-OBJ (PERSONALPRONOMEN "er/sie/es")) (VERB "leihen")

Figure 3: Best Target Structure

Subject Resolution

Applying the decision tree to the morphological analyzed source word sequence results in a leaf containing the subject generation marker <AGEN S> which is assigned to the sentence predicate. During generation this marker will be replaced by its corresponding target expression (PERSONALPRONOMEN ”ich”) and analyzed as the sentence subject.

Generation

The linguistic knowledge summarized in the target structure forms the result of the transfer module. The tree structure is mapped to the following list representation, which forms the input of generation module.

```
(
 (PP (PRAEPOSITION ”für”) (NP (KARDINALZAHL ”4”) (NOMEN ”stunde”)) :CASE AKK)
 (VP (AP+ (ADVERB ”gerne”) (MODALVERB ”mögen” :TENSE KONJUNKTIV-2)))
 (*AKK-OBJ (PERSONALPRONOMEN ”er/sie/es”))
 ( {<AGEN S>} (VERB ”leihen”))
 (AKK-OBJ (NP (LC-EXP (DETERMINATIV ”dieser/diese/dieses”)) (NOMEN ”auto”)))
)
```

During the generation process the list structure is evaluated recursively and the linguistic knowledge is analyzed on word level as well as on sentence level in order to extract the relevant inflection attributes and to update the sentence parts to the appropriate topological fields. Each field is inflected according to the respective attributes, and the order of the topological fields in our sentence model determines the word order of the given translation. In our example the German sentence structure is analyzed as follows.

topo-field	analyzed sentence elements	inflection
subject	((PERSONALPRONOMEN ”ich” (:PERSON 1 :NUMBER SG :CASE NOM)))	”ich”
V-fin	((MODALVERB ”mo∼g” (:TENSE KONJUNKTIV-2 :NUMBER SG :PERSON 1 :MOOD INDIKATIV :VOICE AKTIV)))	”möchte”
VP-adverb	((ADVERB ”gerne”))	”gerne”
akk-object	((DETERMINATIV ”dies” (:GENDER NTR :NUMBER SG :CASE AKK)))	”dieses”
	((NOMEN ”auto” (:NARTICLE OHNE :GENDER NTR :NUMBER SG :CASE AKK :ARTICLE BESTIMMT)))	”Auto”
F-middle	((PRAEPOSITION ”fu∼r” (:CASE AKK)))	”für”
	((KARDINALZAHL ”4” (:GENDER FEM :NUMBER PL :CASE AKK :ARTICLE OHNE :ATTRIBUTIVE-USED-P T)))	”vier”
	((NOMEN ”stunde” (:NARTICLE OHNE :GENDER FEM :NUMBER PL :CASE AKK :ARTICLE OHNE)))	”Stunden”
V-inf	((VERB ”leih” (:NUMBER SG :PERSON 3 :TENSE INFINITIV)))	”leihen”

The concatenation of each topological field in the chosen word order results in the generation output.

”Ich möchte gerne dieses Auto für vier Stunden leihen.”

2 German Peculiarities

In this chapter we want to give a short introduction to the problems arising during the generation of German sentences. The main topics which has to be addressed are the rich inflectional phenomena and the free word order in German.

2.1 Inflectional Phenomena

Based on the functional use within a sentence German words can be classified syntactically into several word categories. According to the morphological characteristics of these word classes we can distinguish two groups: on the one hand the *inflectable word classes*, i.e. words are formed in the grammatical context (conjugation, declension, comparison) and on the other hand *non-inflectable word classes* without morphological modifications. Verbs, nouns, articles and pronouns belong to the inflectable word classes, whereas the group of non-inflectable word classes consists of adverbs, prepositions, particles, conjunctions and interjections.

Verbs can be conjugated in *person* (1,2,3,address), *number* (singular, plural), *tense* (present, imperfect, perfect, past perfect, future), *voice* (active, passive, adjectival passive) and *mood* (indicative, subjunctive, imperative). They can be subclassified in *regular verbs* (regular stem forms are used for the imperfect tense and past participle inflection) and *irregular verbs*, which have special stem forms for the different tenses. According to the functionality within the sentence predicate we distinguish *finite verbs* (tied to personal attribute, with conjugation) and the *infinite verbs* (infinitive, present/past participle).

Nouns have a declension in *number* (singular, plural), *genus* (masculine, feminine, neutral) and *case* (nominative, genitive, dative, accusative) and are characterized by capitalization.

Adjectives are declined in *number*, *genus*, *case* and graduated (positive, comparative, superlative). They can be used either *attributively*, i.e. within a noun phrase in front of the noun, or *predicatively* as a single sentence element.

The class of pronouns can be subclassified as follows, whereby the inflectional attributes are added in brackets (cf. Appendix D, Tabelle 27).

- *personal pronouns* (person, number, case, gender)
- *demonstrative pronouns* (gender, number, case)
- *indefinite pronouns* (gender, number, case)
- *possessive pronouns* (gender, number, case, article)
- *relative pronouns* (person, number, case)
- *reflexive pronouns* (person, number, case)

In combination with a noun, pronouns are used *attributively*. However, in the case of *nominal* use the pronoun substitutes a noun.

In contrast to English, where no relevant case morphology exists, an appropriate handling of the complex morphosyntactic features is crucial for the generation of German words and for the contextual disambiguation of referencing pronouns.

2.2 Free Word Order

One of the main characteristics of the German language is its rather free word order within a sentence. However, the structure of a German clause depends on the position of its finite verb, which can appear in the *first*, *second* or *final* position. The finite verb position plus the non-finite parts of the predicate define the topological structure as listed in Figure 4.

first:	——	*finite verb*	middle field	*non–finite verb*	post field
second:	pre field	*finite verb*	middle field	*non–finite verb*	post field
final:	pre field	——	middle field	*verb–complex*	post field

Figure 4: Topological Fields of the German Sentence Structure

In the *verb-final* case the finite and non-finite part of the sentence predicate forms a compound verb-complex, in which the word order depends on syntactic restrictions.

In contrast to English, the *pre field* is not exclusively reserved for the subject, but can be filled by almost any linguistic constituent depending on the contextual use of this element (topicalization, etc.). However, the *middle field* is in some sense the default field.

The order of the constituents within the middle field (objects, prepositional phrases, etc.) depends on both syntactic ordering principles, e.g. the *unmarked order* (subject, indirect object, object) or complement and adjunct order, and pragmatically based regularities, e.g. focus (cf. [Paul et al. 98]).

2.3 Clause Types

Concerning the functionality, the German sentences are divided in main and subordinated clauses. In contrast to the most subordinated ones, which form the class of *verb-final* sentences, main clauses can either be *verb-second* (statements and complementary questions) or *verb-first* (alternative questions, exclamation clauses and requests).

verb-first:

(1) alternative questions
Haben Sie schon *reserviert?*
have you already reserve
'Have you already made a reservation?'

(2) syntactic imperatives
Kommen Sie so früh wie möglich.
come you as.soon.as.possible
'Please come as soon as possible.'

(3) antecedents of conditionals
Wenn Sie ankommen, *rufen* Sie mich *an.*
when you arrive call you me up
'Call me, when you arrive.'

(4) interjections
Entschuldigen Sie bitte.
excuse you please
'Excuse me.'

verb-second:

(5) main clauses
Ich *habe* ein Taxi *genommen.*
I have a taxi take
'I took a taxi.'

(6) assertion clauses to express questions and orders
Sie *kommen* morgen, nicht wahr?
you arrive tomorrow right?
'You will arrive tomorrow, right?'

(7) constituent questions
Wann *kommen* Sie?
when arrive you
'When will you arrive?'

(8) a. subordinated clauses (sentential complements in *verb-second* order)
Ich habe gehört, Sie *kommen* morgen.
I have hear you arrive tomorrow
'I heard you will arrive tomorrow.'

b. subordinated clauses (first subjunctive indicates indirect speech)
Ich habe gehört, Sie *wird* morgen *kommen.*
I have hear she will tomorrow arrive
'I heard she will arrive tomorrow.'

verb-final: subordinated clauses, introduced by:

(9) a conjunction
Ich rufe Sie an, *nachdem* ich *angekommen bin.*
I call you after I arrive be
'I will call you after my arrival.'

(10) a complementizer
Ich vermute, *daß* ich später *ankomme.*
I think that I later arrive
'I think, that I will arrive later.'

(11) subordinated clauses, introduced by an interrogative item
Ich weiß nicht, *wann* ich *ankommen werde.*
I know not when I arrive will
'I don't know, when I will arrive.'

(12) subordinated clauses, introduced by a relative pronoun
Ich nehme den Bus, *der* zum Flughafen *fährt.*
I take the bus that to airport go
'I will take the bus, that is going to the airport.'

3 German Word Model

Due to the rich inflectional phenomena of German we have chosen a stem representation for our word model, whereby a surface word is classified by its stem lexem and the appropriate suffix, depending on the inflectional attributes of the word. This compact representation scheme reduces the size of our generation dictionary by a factor of 7 compared to a fullform word representation (cf. Chapter 5.2).

However, the additional computational costs for analyzing/generating the inflectional attributes of a surface word requires an efficient handling of the word analysis. Within the German generation module of TDMT we make use of the morphological package MORPHIX[5], which handles all inflectional phenomena of German. This classification-based approach considers morphological regularities of the input language as the basis for the definition of a fine-grained, word-class specific classification, i.e. words with the same morphological behavior are grouped together in classes. Additional it uses morphosyntactic features (phonological properties) in refining the class hierarchy.

In addition to the *fullform-lexicon* for storing the non-inflecting word-classes (e.g. adverbs), the *stem-lexicon* contains information about the classification of each word-stem, and the *inflectional allomorph lexicon* (IAL) relates each inflectional morph to all its possible combinations of morphosyntactic information. Each entry in the IAL is an n–ary tree, of which the nodes describe the classes and the leaves contain the appropriate inflectional information (cf. Figure 6).

These lexicons form the central knowledge sources of our word model and the morphological analysis/generation can be performed by means of simple operations on n–ary trees, which allows

[5] MORPHIX was developed within the Special Collaborative Program on AI and Knowledge-Based Systems (SFB 314), project N1 (XTRA) of the German Science Foundation (DFG) by Wolfgang Finkler and Günther Neumann [Finkler & Neumann 88].

an efficient implementation and yields in an excellent run time behavior, despite the complexity of the German inflection.

3.1 Classification

According to the capability of inflection the word classes can be divided into two groups. The non-inflecting word classes (e.g. prepositions, conjunctions, adverbs) are stored in the *fullform-lexicon* and only a lexicon access is necessary. As the non-inflecting word-classes are closed and domain-independent, the word-form lexicon belongs to the fixed lexical knowledge of MORPHIX.

In contrast, the inflecting word classes (e.g. verbs, nouns, adjectives) will be classified after their lexem as well as the inflectional word suffix and represented in the *stem-lexicon.*

3.1.1 Lexems

The general schema for the classification of the inflectable words is as follows:

part of speech
↓
inflectional regularities
↓
irregular stems
↓
word-end phenomena

However, this general schema varies for each part of speech. The specific classification attributes for each part of speech are listed below.

Noun The classification of nouns in MORPHIX is adopted from [Schott 72]. Each noun has the following characteristics:

canonical stem [stamm]
Nominative singular or nominative plural (for plural nouns), used as the key for the lexicon entry.

gender [gender]
MAS (masculine), FEM (feminine), NTR (neuter).

umlaut [uml]
For nouns, which transform a stem-vowel to an umlaut in the plural form.

singular class [ksg]
Differentiation of word suffixes for the genitive or dative singular (cf. Table 2a).

plural class [kpl]
Differentiation of word suffixes for the nominative and dative plural (cf. Table 2b).

Example:

haus : ((wortart . NOMEN) (genus . NTR) (uml . T) (ksg . 3) (kpl . 9))
ansatz : ((wortart . NOMEN) (genus . MAS)(uml . T) (ksg . 3) (kpl . 7))
atlas : ((wortart . NOMEN) (genus . MAS) (ksg . 5) (plural . "atlanten")
atlanten : ((stamm "atlas" . plural) (wortart . NOMEN) (genus . MAS) (kpl . 1)

Table 2: Noun Classification

ksg	GEN	DAT	example
1	–	–	Frau
2	s	–	Auge
3	es	–/e	Hals
4	s/es	–/e	Wort
5	ses	–/se	Kürbis
6	ens	–/en	Herz
7	ns	n	Name
8	n	n	Hase
9	en	en	Mensch

(a) singular classes

kpl	NOM	DAT	example
01	–	–	Wagen
02	s	s	Auto
03	n	n	Bote
04	en	en	Student
05	nen	nen	Lehrerin
06	–	n	Vater
07	e	en	Ball
08	se	sen	Geheimnis
09	er	ern	Haus
10	ien	ien	Prinzip
11	sen	sen	Kirmes
12	ten	ten	Bau
13	te	ten	Klima

(b) plural classes

Verb The classification of verbs in MORPHIX is adopted from [Busemann 83]. Each verb has the following characteristics:

canonical stem [stamm]
The reduced infinitive form (suffixes ”en” or ”n” split off), used as the key for the lexicon entry.

irregular stem
The inflection of irregular verbs uses different stems (changes of stem vowels or consonants). For efficiency reasons the irregular stems are used as a key for the stem-lexicon with a link to its canonical stem (no algorithmic drawback from the irregular stem to the canonical one). The different stems are grouped in 4 basic verb forms (VGFA, VGFB, VGFC, VGFD), the participle perfect stem (PPRF) and 4 special stem groups (SOND1-4) for the auxiliary verb ”sein” (cf. Table 3).

Table 3: Irregular Verb Stem Classification

stem classification	used for	example
VGFB	2./3. singular present	triff
VGFC	imperfect	traf
VGFD	subjunctive 2	träfe
VGFA+	canonical stem with deleted ”e” for verbs with suffixes ”ern” or ”eln”	handl
VGFA	other conjugations	treff
PPRF	participle perfect	troff
SOND1	1. singular present (sein)	bin
SOND2	2. singular present (sein)	bist
SOND3	3. singular present (sein)	ist
SOND4	1./3. plural present (sein)	sind

stem subclasses
In order to avoid algorithmic computations of the final stem position, different subclasses of the irregular stem classification will be defined (cf. Table 4).

Table 4: Subclass Classification of Irregular Verbs

subclass	final stem position	example
A2	”d”, ”t”, ”j”	arbeit
A2	”m”, ”n” with preceding consonant, others than ”l”, ”m”, ”n”, ”r” or ”< *vowel* >+h”	atm
A3	modal verb ”sollen”	soll
A4	”s”, ”ß”, ”z”, ”x”	ras
A5	”el”	sammel
A6	”er”	erneuer
A7	”sein” or ”tun”	tu
A1	others	rasier

(a) subclasses of VGFA

subclass	2. sg	3. sg	example
B1	st	t	gib
B2	–	–	birst
B3	st	–	brät
B4	t	t	iß
B5	st	d	wir
B6	t	–	muß

(b) subclasses of VGFB

subclass	final stem position	example
C2	”s”, ”z”, ”x”	wuchs
C3	”d”, ”t”	trat
C4	”e”	mochte
C1	others	fuhr

(b) subclasses of VGFC

verb type [vtyp]

Differentiation of the combinations of the stems used for the conjugation (cf. Table 5).

verb prefix [vrbz]

For some verbs the prefix can be splitted off. Therefore it is sufficient to know the length of the prefix. The classification of such prefix-verbs are put down to the classification of the basic verb, i.e. the part of the prefix-verb without the prefix.

auxiliary verb [paux]

For the compound verb tenses we have to know, which auxiliary verb (”sein” or ”haben”) has to be used for the inflection.

participle perfect

The form of the verb used for the participle perfect inflection.

Example:

```
rasier  : ((wortart . VERB) (vtyp . 1) (VGFA . A1)
           (paux . ”hab”) (PPRF . ”rasiert”))
treff   : ((wortart . VERB) (vtyp . 7) (VGFA . A1)
           (VGFB B1 . ”triff”) (VGFC C1 . ”traf”) (VGFD . ”träfe”)
           (paux . ”hab”) (PPRF . ”getroffen”))
triff   : ((stamm ”treff” . VGFB)
antreff : ((wortart . VERB) (vrbz 2 . T) (paux . ”hab”) (PPRF . ”angetroffen”))
nenn    : ((wortart . VERB) (vtyp . 2) (VGFA . A1) (VGFC C4 . ”nannte”)
           (paux . ”hab”) (PPRF . ”genannt”))
```

Table 5: Verb Classification

class	1. sg present	2./3. sg present	imperative singular	imperfect	subjunctive 2	example
1	VGFA	VGFA	VGFA	VGFA	VGFA	hol
2	VGFA	VGFA	VGFA	VGFC	VGFA	kenn
3	VGFA	VGFA	VGFA	VGFC	VGFC	bleib
4	VGFA	VGFA	VGFA	VGFC	VGFD	beginn
5	VGFA	VGFB	VGFA	VGFC	VGFC	blas
6	VGFA	VGFB	VGFA	VGFC	VGFD	fahr
7	VGFA	VGFB	VGFB	VGFC	VGFD	geb
8	VGFB	VGFB	VGFA	VGFA	VGFA	woll
9	VGFB	VGFB	VGFA	VGFC	VGFA	könn
10	VGFB	VGFB	VGFA	VGFC	VGFD	wiss
11	SOND1	SOND2/3	VGFA	VGFC	VGFD	sei

Adjective The classification of adjectives in MORPHIX is adopted from [Helbig & Buscha 72]. Each adjective has the following characteristics:

canonical stem [stamm]
Basic form of the adjective without a suffix, used as the key for the lexicon entry.

adjective class [main]
Adjectives can be used attributively as well as predicatively (A), or either only attributively (B) or only predicatively (C). These 3 adjective classes can be sub-classified after their possibility to be inflected (infl) and graduated (grad) as described in Table 6a.

comparison class [komp]
For those adjectives, which can be graduated (subclasses A1 and B1) we need the information how to inflect the superlative. This can be done by appending either the suffix "st" or the suffix "est" to the canonical stem. Further classification takes into account the stem transformation because of an umlaut and an elision or the use of irregular stem forms (cf. Table 6b).

umlaut [uml]
For adjectives, which transform a stem vowel to an umlaut for the inflection of the comparative and superlative forms (cf. Table 6c).

elision [elision]
For adjectives, which undergo an elision (i.e. the deletion of a vowel) in the final stem position during the inflection (cf. Table 6d).

stem-end 'e' predicative [pred-e]
For predicative used adjectives with "e" in the final stem position.

Table 6: Adjective Classification

class	description	example
A1	infl + grad	wichtig
A2	only infl	ledig
A3	no infl, no grad	lila
B1	infl + grad	vorder
B2	only infl	vorig
B3	no infl, no grad	keinerlei
C1	no infl, no grad	schade

(a) adjective classes

class	superlative	example
1	append ”st”	jung
2	append ”est”	erfolglos
3	irregular stem (KOM,SUP)	gut
4	irregular stem (POS,KOM,SUP)	hoch

(b) comparison classes

class	umlaut	example
0	no	dunkel
1	necessary	alt
2	possible	schmal

(c) umlaut classes

class	elision	example
0	no	alt
1	necessary	dunkel
2	possible	heiter

(d) elision classes

Example:

edel : ((wortart . ADJEKTIV) (main . A1) (komp . st) (elision . nec))
edl : ((stamm ”edel” . elision))
bös : ((wortart . ADJEKTIV) (main . A1) (komp . est) (pred-e . T))
zusätzlich : ((wortart . ADJEKTIV) (main . A2))
hoch : ((wortart . ADJEKTIV) (main . A1) (komp . irreg–pos)
(pos . ”hoh”) (kom . ”höher”) (sup . ”höchst”))

Possessive Pronoun

Each possessive pronoun has the following characteristics:

canonical stem [stamm]
the not inflectable form, used as the key for the lexicon entry.

elision [elision]
For possessive pronouns, which undergo an elision (i.e. the deletion of a vowel) in the final stem position during the inflection.

Example:

dein : ((wortart . POSSESSIVPRONOMEN))
unser : ((wortart . POSSESSIVPRONOMEN) (elision . pos))
unsr : ((stamm ”unser” . elision))

Article

Each article has the following characteristics:

numerus [det-num]
Articles can be used as well in singular form as plural form (sg+pl) or only in singular form (sg) or only in plural form (pl).

Example:

kein : ((wortart.DETERMINATIV-INDEF) (det-num.sg+pl))
dies : ((wortart.DETERMINATIV) (det-num.sg+pl))
jed : ((wortart.DETERMINATIV) (det-num.sg))
all : ((wortart.DETERMINATIV) (det-num.pl))

Ordinal

Each ordinal number has the following characteristics:

canonical stem [stamm]
the affiliated cardinal stem, used as the key for the lexicon entry.

generate from cardinal [ending]
For ordinal, which forms the inflection by appending a suffix to its stem (cf. Table 7a).

ordinal stem [ordinal]
For ordinal, which uses an irregular stem for the inflection (cf. Table 7b).

Table 7: Ordinal Classification

class	suffix	example
t	"t"	zwei
st	"st"	zwanzig
no	–	acht

(a) append suffix

class	"zu" + ordinal	example
ordinal	possible	dritt
ordinal-b	not possible	erst

(b) irregular stem

Example:

zwei : ((wortart.KARDINALZAHL) (ending.T))
drei : ((wortart.KARDINALZAHL) (ordinal."dritt"))
dritt : ((stamm "drei".ordinal))

3.1.2 Suffixes

In order to analyze words which are not included in the fullform-lexicon, a decomposition of the word into a candidate stem and the corresponding prefix and suffix has to be done. Therefore we need some information about the longest possible inflection suffix of the input. In MORPHIX the classification of the suffixes are ordered hierarchically and stored in the IAL lexicon. In Figure 5 the hierarchy for the suffix "N" is listed.

Each inflection suffix in the suffix tree (cf. Appendix C, page 60) is assigned to an n–ary tree, that represents all possible part-of-speech classifications (*IAL tree*) and contains the morphosyntactic information in the leaves of the tree (cf. Figure 6).

Thus, given the segmentation of a surface word in its stem and suffix representation we can analyze the inflectional attribute information by finding a path in the appropriate suffix tree, which fulfills the classification attributes of the stem.

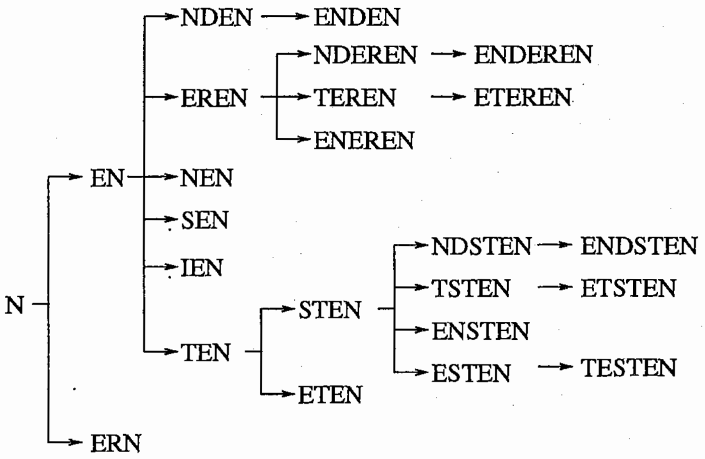


Figure 5: Suffix Tree "N"

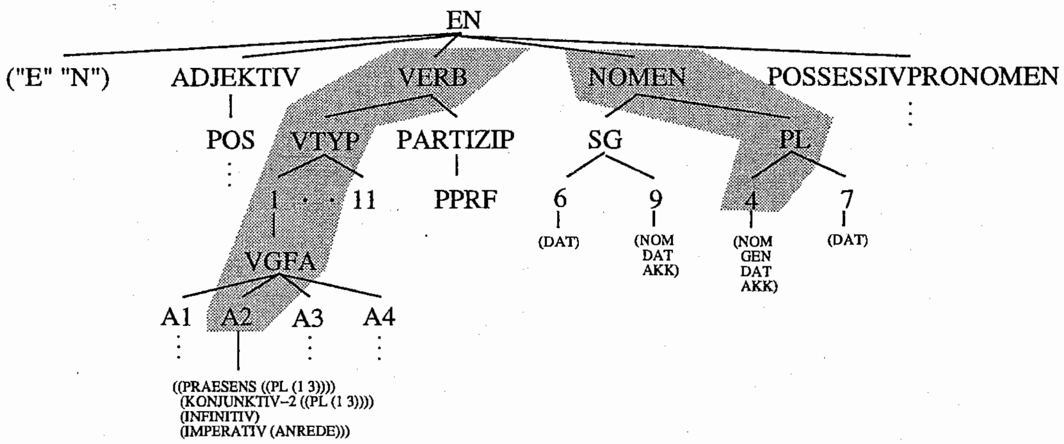


Figure 6: IAL Tree for Suffix "EN"

3.2 Encoding

The word classification information, i.e. the morphological characteristics of the German lexemes, is encoded using 6-digit numbers (cf. Appendix C, page 59). Entries marked with **l-s** will be added to the stem-lexicon and the **l-f** entries to the fullform-lexicon.

The classification information of the current generation dictionary entries are extracted automatically from the CELEX lexical database (cf. [Piepenbrock 95]). This morphological and syntactic information is mapped into the digital representation scheme of MORPHIX (cf. Chapter 5.2).

However, if a word is not in the database the lexicon can be interactively enhanced by means of a clarification dialog (cf. Appendix C, page 61). For portability purposes the clarification dialog of the MORPHIX package is implemented as a line-oriented dialog (cf. Figure 7 for an overview of all masks of the clarification dialog).

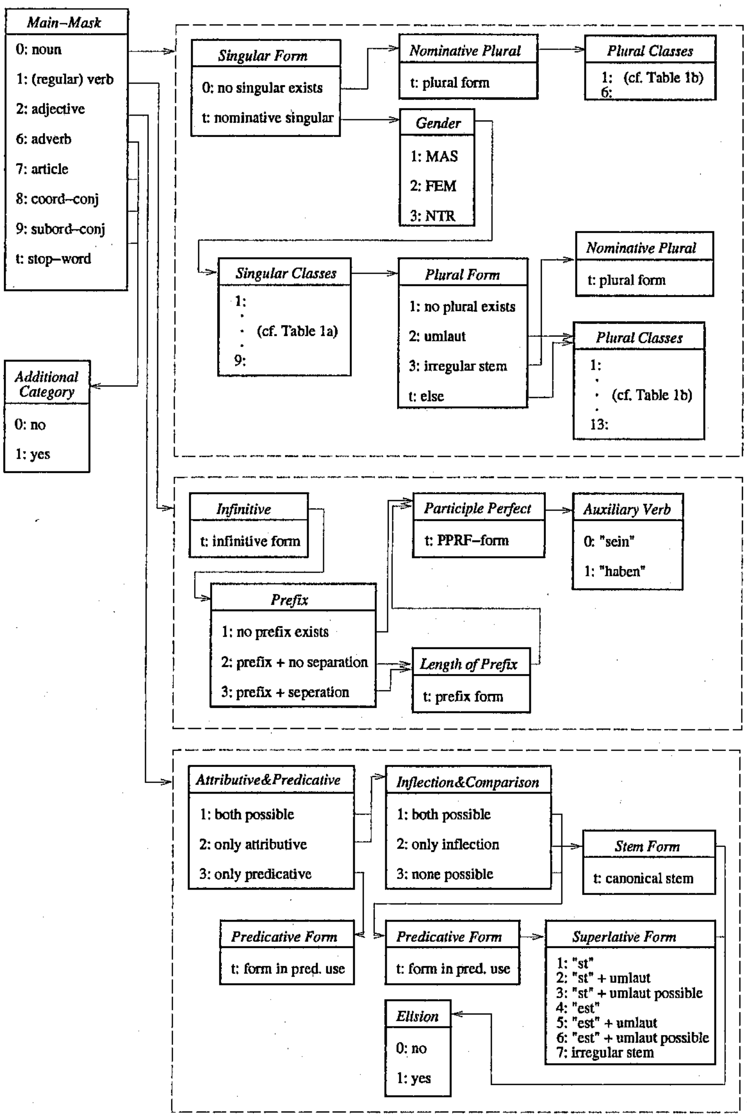


Figure 7: Clarification Dialog

Several questions about the part-of-speech, conjugation forms, etc. have to be answered by the user. Thus only some linguistic knowledge, e.g. whether an umlaut transformation for adjectives occurs in the comparison form or not, but no knowledge about the word classification itself is necessary to enhance the dictionary. The MORPHIX system also uses existing lexicon entries and relations from the stem to their subclassifications in order to save the user from carrying out the entire classification by hand.

Because several part-of-speeches (e.g. personal pronouns, irregular verbs) are restricted and entirely covered by the fixed lexical knowledge of MORPHIX, not all part-of-speeches have to be handled by the clarification dialog. The categories,which can be added to the lexicon, are common nouns, regular verbs, adjectives, adverbs, conjunctions and stop-words (words without any morphological analysis, used for proper nouns).

3.3 Segmentation

In order to find all correct segmentations for a given input word, it is necessary to split a complex suffix into its parts, i.e. taking into account substrings of the longest possible suffix, and to handle all possible prefix-stem-suffix combinations. A segmentation is *correct*, if the stem can be found in the stem-lexicon and if the concatenation of stem and inflectional morpheme represents a grammatically well-formed surface-word. The segmentation algorithm is described in Figure 8.

For illustration we use the input "rasten". In the table below all possible segmentations are listed.

"rasten"

(9) ⇓

"ra" + "sten" **(11)** ⇒

stem – suffix
ra – sten
ras – ten
rast – en
raste – n
rasten – ∅

(13) ⇓

stem	classification of stem in the stem-lexicon	search path
ra	–	–
ras	((wortart . VERB) (vtyp . 1) (VGFA . A4))	VERB–vtype–1–VGFA–A4
rast	((wortart . VERB) (vtyp . 1) (VGFA . A2))	VERB–vtype–1–VGFA–A2
	((wortart . NOMEN) (genus . FEM) (ksg . 0) (kpl . 4)	NOMEN–SG–0 NOMEN–PL–4
raste	((wortart . NOMEN) (genus . FEM) (ksg . 0) (kpl . 3)	NOMEN–SG–0 NOMEN–PL–3
rasten	–	–

(14) ⇓

IAL trees for suffixes "ten", "en", "n" (cf. Figure 6)

(16) ⇓

result of morphological analysis	
("ras"	(WORTART VERB) (FLEXION ((IMPERFEKT ((SG (ANREDE)) (PL (1 3 ANREDE)))) (KONJUNKTIV-2 ((SG (ANREDE)) (PL (1 3 ANREDE))))))
("rast"	(WORTART NOMEN) (FLEXION ((FEM ((PL (NOM GEN DAT AKK))))))
("rast"	(WORTART VERB) (FLEXION ((PRAESENS ((SG (ANREDE)) (PL (1 3 ANREDE)))) (KONJUNKTIV-1 ((SG (ANREDE)) (PL (1 3 ANREDE))) (INFINITIV) (IMPERATIV (ANREDE))))
("raste"	(WORTART NOMEN) (FLEXION ((FEM ((PL (NOM GEN DAT AKK))))))))

In our example we could analyze four ambiguous segmentations of the input string "rasten". The stem "ras" is analyzed as the imperfect form of the verb "rasen" (*to rage*), whereas the stem "rast"[6] refers either to the infinitive verb "rasten" (*to rest*) or the noun "rast" (*rest*) in the plural

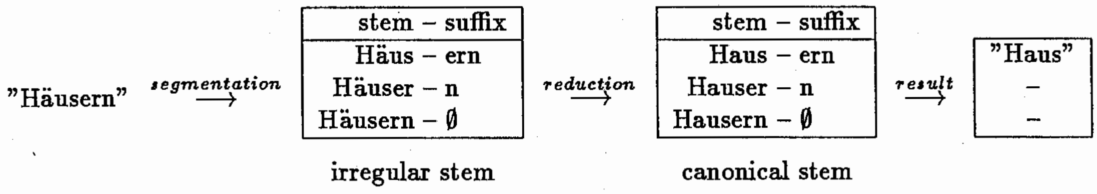
[6] The search path for the IAL tree of these segmentations are marked in Figure 6

form. The fourth segmentation represents the plural form of the noun "raste" (*notch*).

```
proc Segmentation( $Word$ ) ;
begin
  if $Word \in *FullformLexicon*$ then
    $CorrectSegments \leftarrow (\emptyset \,.\, Word \,.\, \emptyset)$ ;
  else
    $CorrectSegments \leftarrow \emptyset$ ;
    $Prefix \leftarrow$ Get–Longest–Prefix($Word$) ;
    $Word \leftarrow$ Split–Off($Prefix, Word$) ;
    $Suffix \leftarrow$ Get–Longest–Suffix($Word$) ;
    while $Suffix$ do
      $Stem \leftarrow$ Split–Off($Suffix, Word$) ;
      if $Stem \in *StemLexicon*$ then
        $SearchPaths \leftarrow$ Get–Classification($Stem$) ;
        $NAryTree \leftarrow$ Get–IAL–Tree($Suffix$) ;
        forall $Path \in SearchPaths$ do
          if Match($Path, NAryTree$) and Leaf–Reached–P($Path$) then
            $CorrectSegments \leftarrow CorrectSegments \cup \{(Prefix \,.\, Stem \,.\, \mathrm{Leaf}(Path))\}$ ;
          fi ;
        od ;
      fi ;
      $Suffix \leftarrow$ Delete–First–Character($Suffix$) ;
    od ;
  fi ;
  return( $CorrectSegments$ ) ;
end ;
```

Figure 8: Segmentation Algorithm

Because of the use of irregular stems and some language-specific phenomena, e.g. the german umlaut, the segmentation process may not yield the canonical stem. The first point can be handled by storing the irregular stems in the stem lexicon and add a link to its canonical stem. During the inflection in German, e.g. of nouns, the use of an umlaut transformation in the stem plus the inflectional suffix occurs very often. In order to limit the number of entries in the stem-lexicon these umlaut phenomena will be reduced and only the canonical stem has to be stored in the stem-lexicon. Thus, given the German word *Haus* the dative plural form *Häusern* will be segmented as follows, whereby the irregular stem will be reduced to the canonical one.



3.4 Inflection

The word classification in MORPHIX can also be used for the generation of German words by reversing the word segmentation algorithm described in Chapter 3.3. Given a word stem plus some inflection attributes we use the suffix inflection information coded in the IAL tree to generate the surface word as a concatenation of the stem and the appropriate suffix as listed in Figure 9.

```
(1)  proc Generation( $Prefix, Stem, InflectionAttributes$ ) ;
(2)  begin
(3)     if $Stem \in *StemLexicon*$ then
(4)        $SearchPath \leftarrow$ Get-Classification($Stem$) ;
(5)        $SurfaceStem \leftarrow$ Get-Surface-Form($SearchPath$) ;
(6)        $NAryTree \leftarrow$ Get-IAL-Tree-With-Leaf($InflectionAttributes$) ;
(7)        forall $Path \in SearchPaths$ do
(8)           if Match($Path, NAryTree$) then
(9)              $SurfaceSuffix \leftarrow$ Get-Root($Path$) ;
(10)          fi ;
(11)       od ;
(12)       return( Concatenate($Prefix, SurfaceStem, SurfaceSuffix$ ) ;
(13)    else
(14)       if $Stem \in *FullformLexicon*$ then
(15)          return( $Stem$ ) ;
(16)       else
(17)          ClarificationDialog($Stem$) ;
(18)       fi ;
(19)    fi ;
(20) end ;
```

Figure 9: Generation Algorithm

First the surface stem will be generate according to the classification information of the canonical stem found in the stem-lexicon. In order to handle irregular stem forms, the additional links of the canonical stems to their irregular stems will be used. If an umlaut is marked in the stem classification, the umlaut has to be re-transformed (e.g. "Haus" ⟹ "Häus"). Also the characteristics of a stem consonant or vowel change or elision has to be handled in order to get the correct surface stem.

In the next step the correct suffix will be searched and appended to the surface stem, yielding in the correct inflection word form. The generation algorithm is described in Figure 9.

As an example we describe the generation of the canonical stem "rast" given the inflection attributes VERB (part-of-speech), PRAESENS (tense), PL (number) and 1 (person):

"rast" + (VERB PRAESENS PL 1)

(4) ⇓

stem	classification of stem in the stem-lexicon	search path
rast	((wortart.VERB)(vtyp.1)(VGFA.A2))	VERB–vtype–1–VGFA–A2

(5) ⇓

Surface Stem: **"rast"**

(6) ⇓

IAL tree with leaf matching [VERB PRAESENS PL 1] (cf. Figure 6)

(9) ⇓

Surface Suffix: **"en"**

(12) ⇓

"rasten"

If no information was found in the stem-lexicon a clarification dialog can be invoked in order to augment the lexicon (cf. Chapter 3.2).

4 German Sentence Model

In TDMT the structure of the input sentence is analyzed by identifying functional words and explicitly inserted bigram markers as constituent boundaries. The hierarchical structure of the input sentence is determined by the application of lexical transfer rules on various linguistic levels. For each source structure constituent a corresponding target constituent is selected in the context of the closest example of the training data. Thus the attachment of subconstituents, e.g. adnominal versus adverbial usage of prepositional phrases, is based on the knowledge of our example database. However, the constituent boundary parsing method cannot achieve a global ordering of the main constituents of the target structure, because the lexical transfer rules are applied locally without any knowledge of the remaining constituents specified in the source structure.

Due to the lack of a grammar formalism, we cannot determine the word order of the sentence by using formal ordering rules. Thus we have to define a sentence model, which uses the linguistic knowledge of the target structure for determining the order of the analyzed sentence constituents in an appropriate way.

In the German generation module a sentence consists of several topological fields representing the main constituents of a German clause, e.g. subject, objects, time and locative expressions. Therefore the coarse sentence model, introduced in Chapter 2.2 is refined by subdividing the middle field and defining an order of its constituents.

In order to handle the large number of combinational possibilities due to the rather free word in German, we define a *default model* for each of the possible finite verb positions as listed in Table 8.

The chosen word order, which is represented by the linear composition of the topological fields, is based on syntactic ordering principles and the analysis of our training data.

Table 8: Topological Fields in TDMT

:first	:second	:final
intro	intro	intro
pre field	pre field	pre field
finite verb	subject	reflexive pronoun
reflexive pronoun	finite verb	subject
subject	reflexive pronoun	dative object
dative object	dative object	verb phrase (adverbial)
negation	verb phrase (adverbial)	time
verb phrase (adverbial)	time	negation
time	place	place
place	nominative object	nominative object
nominative object	accusative object	accusative object
accusative object	genitive object	genitive object
genitive object	negation	middle field
middle field	middle field	prepositional phrase
prepositional phrase	prepositional phrase	verb phrase (non verbal)
verb phrase (non verbal)	verb phrase (non verbal)	infinite verb
infinite verb	infinite verb	finite verb
post field	post field	post field

Besides the specification of the topology the sentence model contains additional features concerning the syntactic characteristics of the sentence (cf. Appendix D, Table 25). Depending on the clause type, specific attributes (tense, mood, person) are required for the sentence predicate. Additionally, an appropriate punctuation sign has to be defined for each sentence type. In the case of subordinated sentences, specific introductory phrases are required according to their grammatical usage.

For each sentence type a sentence model is derived from the default models, whereby the inherited features can be overridden by adopting sentence type specific characteristics, e.g. in the case of a subordinated sentence of type INH-S-ZU (infinitive sentence plus the prepositional verb addition "zu") the sentence model is derived form the *:final* default model forcing the infinitive tense and dropping the subject field, because a subject is not possible in a INH-S-ZU subclause.

Thus the general topology of a subordinated sentence has to be redefined in order to fulfill the syntactic constraints of the respective sentence type.

For each sentence model a default order of its topological fields is defined. However, this static order has to be adopted dynamically according to the contents of the constituents specified in the generation input.

For example, the specification of a personal pronoun as the sentence subject or as the accusative object influences the word order in a German sentence. In the case of a pronominal accusative object, the respective topological field has to be shifted in front of the dative object in order to achieve a natural translation. The specification of a pronominal subject leads to similar modification concerning the positioning of successive reflexive pronoun and subject fields. A pronominal subject has to be generated in front of the reflexive pronoun.

5 Generation Module

The word and sentence models described in Chapter 3 and 4 form the main knowledge sources for the analysis of the linguistic constituents specified in the transfer result. The generation input is a list structure consisting of target expressions which represents the best translation of the analyzed source patterns in the context of our example sentences. According to the structure of the target tree, simple constituents (entries of the transfer dictionary) are combined to more complex phrases resulting in the main constituents (*top-level* constituents) of the target sentence.

In order to generate an appropriate German sentence the specified constituents have to be analyzed on phrase level as well as on sentence level. On phrase level the morphological attributes of the word translations have to be constraint according to the grammatical function of the specified linguistic markers, e.g. the sentence subject marked by SUB requires the nominative case for its subcomponents. On sentence level the linguistic markers of the top-level constituents determine the appropriate topological field for the generation of the analyzed constituent and thus the word order of the target sentence.

However, before explaining the generation process in detail, we have to introduce the terminology used for the specification of the German target expressions (cf. Chapter 5.1). The generation dictionary contains the morphological, syntactic and phonetic information of each German word occurring in TDMT and it is created automatically from a lexical database (cf. Chapter 5.2). The analysis of the transfer result consists of the following three steps. A preprocess adjusts the syntax of the target specifications for the analysis (cf. Chapter 5.3). The type of the target sentence is analyzed and the respective sentence model is used for the analysis of the target constituents (cf. Chapter 5.4). The inflection of the analysis results is described in Chapter 5.5. The generation output of complex sentences (punctuation marks, etc.) are adjusted in a postprocess (cf. Chapter 5.6) and the resulting translation string forms the output of the generation module.

Finally we will give an outline of the German interface to the synthesis module CHATR, which contains not only the complete morphological and syntactic, but also the phonetic information of the generated translation.

5.1 Terminology

In general, a target expression is a list which consists of a marker specifying the type of the expression, one or more subexpressions and an optional list of inflection attributes:

$$\text{exp} = (\mathit{type}\ \{\ \mathit{exp}_{sub}\ \}+\ \{\ (\ \{\ \mathit{attr}\ \}+\ )\ \}^*\ )$$

{}* zero or more elements
{}+ one or more elements

Depending on the syntactic use, we distinguish four groups of target expressions plus some additional generation markers.

Words and Numbers

Each string in the generation input is considered as one or more German words separated by white spaces. The spelling of German words contains special diacritic characters (ä,ö,ü,ß;é, etc.). In order to be independent from the chosen font coding, the diacritics have to be transformed into an internal representation. The conventional transcription of German *umlauts* uses the 'e' vowel as a suffix to the umlaut vowel (ä=ae,ö=oe,ü=ue) and 'ss' for 'ß'. However, this representation leads to ambiguity during retranscription of the generation output ("Mauer" (*wall*) → no transformation of "ue" to "ü"). Therefore we have chosen an unambiguous internal representation for the diacritic transcription (cf. Chapter D, page 78).

A single word is represented by its canonical stem (cf. Chapter 3). Based on the knowledge of our word model, each word is associated with a set of morphological attributes specifying its inflectional characteristics. However, due to the internal stem representation a simple word can have ambiguous morphological characteristics.

Numbers are specified as a sequence of digits in the generation input. They have to be transformed to their corresponding cardinal words. Based on the classification of the basic cardinal and ordinal numbers, we define the word classification of numbers, not contained in the generation dictionary, during the analysis process.

Part-of-Speech

The disambiguation of the morphological word classification is done by assigning a part-of-speech tag to each word. The set of part-of-speech tags used in the current system is listed in Appendix D, Table 22. The format of a part-of-speech expression is as follows:

$$(\mathit{Part\text{-}of\text{-}Speech\text{-}Tag}\ \mathit{Word}\ \{\ (\ \mathit{attr\ value}\ \ldots\ )\ \}^*\ )$$

If an attribute-value list is specified for the part-of-speech, the values are used to constraint the inflectional attributes of this target expression.

GenMarker

The generation markers summarized in Appendix D, Table 24 consist of tags, encapsulated in {}-brackets, and are used either at sentence-level (negation and subject marker) or at phrase-level (article and pronoun marker) in order to force specific constraints to the target constituents.

Part-of-Sentence

According to the usage (grammatical, derivative, functional, topological, inflectional) we define various part-of-sentences which group the part-of-speeches to complex constituents. The format is as follows:

(*Part-of-Sentence-Tag* { *Part-of-Speech* | *Word* | *GenMarker* }+ { (*attr value* ...) }*)

Besides the restriction of the inflectional attribute by the optional attribute-value list, the part-of-sentence itself can force specific constraints to the enclosed elements, e.g. the *case* attribute of the respective object types. In Appendix D, Table 23 we give an overview of the defined part-of-sentences in the current system and a short explanation of their use.

Defaults

Part-of-speech and part-of-sentence tags marked with a leading '*' are called *defaults.* The default constituents are analyzed in analogy to the corresponding non-default constituents. However, whenever a non-default constituent is specified in the generation input, its analysis will overwrite the default elements. For example, a *SUB constituent is analyzed as the target subject provided that no SUB constituent is contained in the transfer result.

Sentence

A simple sentence is a sequence of target expressions consisting of part-of-sentences, part-of-speeches, generation markers and optional sentence type tags (if the sentence type is not explicitly specified, it defaults to a statement). In the case of a complex sentence, all simple sentences contained in the main sentence are marked either with the coordinated sentence marker SATZREIHE or the subordinated sentence marker SATZGEFUEGE. The sentence marker FORCE-S forces the specified sentence type to all subordinated clauses.

The syntax of the target expressions, specified in the transfer result, is listed in Table 9.

5.2 Generation Dictionary

The generation dictionary contains the definitions of the syntactic, morphological and phonetic properties of the German words used in the current system. The format of each entry in the dictionary is as follows:

(*stem* *phon* *stress* *morph*)

stem	word stem (string)
phon	word phoneme (string of phonemes with phon and syllable boundaries)
stress	word stress (string of '0' and '1' marking the stress of the corresponding syllables)
morph	morphological word classification (cf. Chapter 3)

The knowledge about German words used in the current system is extracted from the CELEX lexical database (cf. [Piepenbrock 95]). Apart from orthographic features, the CELEX database comprises representations of the phonological, morphological and syntactic properties of about 52000 word stems (lemmatas) with 365530 corresponding wordforms.

Table 9: Syntax of the Transfer Result

Sen	::=	NIL | NotNilSen
NotNilSen	::=	MainSen | SubordSen | CoordClause | SubordClause
MainSen	::=	({MainSenElem}+ {InfAttr}*)
SubordSen	::=	({SubordSenElem}+ {InfAttr}*)
CoordClause	::=	(SATZREIHE | FORCE-S {MainSenElem}+ {InfAttr}*)
SubordClause	::=	(SATZGEFUEGE {SubordSenElem}+ {InfAttr}*)
MainSenElem	::=	MainSenTag | SenElem | NotNilSen
MainSenTag	::=	<cf. Appendix D, Table 25, main clauses>
SubordSenElem	::=	SubordSenTag | SenElem | NotNilSen
SubordSenTag	::=	<cf. Appendix D, Table 25, subordinated clauses>
SenElem	::=	SubordClause | PartOfSen | PartOfSp | WordAtom | GenMarker
PartOfSen	::=	({PartOfSenTag | DefPartOfSenTag} {SenElem}+ {InfAttr}*)
PartOfSenTag	::=	<cf. Appendix D, Table 23>
DefPartOfSenTag	::=	*PartOfSenTag
PartOfSp	::=	({PartOfSpTag | DefPartOfSpTag } WordAtom {InfAttr}*)
PartOfSpTag	::=	<cf. Appendix D, Table 22>
DefPartOfSpTag	::=	*PartOfSpTag
GenMarker	::=	<cf. Appendix D, Table 24>
WordAtom	::=	Word | Number
Word	::=	"String_1 String_2 ..."
String	::=	"..."
Number	::=	1 | 2 | ...
InfAttr	::=	AttrKey AttrValue
AttrKey	::=	<cf. Appendix D, Table 27>
AttrValue	::=	<cf. Appendix D, Table 26>

We are using the lemmata database files to extract the complete word classification required by the chosen word model as listed in Figure 10.

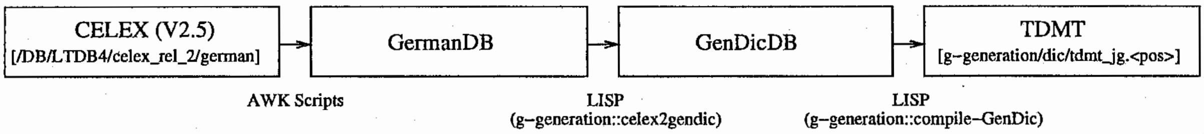

CELEX (V2.5)
[/DB/LTDB4/celex_rel_2/german]
GermanDB
GenDicDB
TDMT
[g-generation/dic/tdmt_jg.<pos>]
AWK Scripts
LISP
(g-generation::celex2gendic)
LISP
(g-generation::compile-GenDic)

Figure 10: Creation of the Generation Dictionary

The CELEX database is divided into several subpartitions according to the represented properties. In a first step the relevant information (stem, phoneme, stress, morphological properties) of each word is extracted from the respective database files using AWK scripts provided by CELEX for the join operation. Because the phoneme transcription in CELEX differs from the one used in the CHATR module, the stem phonemes are mapped to CHATR's phone set before joining the complete word information.

The modified database (GermanDB) is then converted to the format of our generation dictionary, whereby the morphological information of each word is used to define the word classification scheme of our word model described in Chapter 3.

However, not all words of our travel arrangement task are contained in the CELEX database. Thus we first added all unknown words (mainly proper nouns) by hand. Proper noun parts, which are not regular German words, e.g. the temple name *Kinkakuji* in the noun phrase "*Kinkakuji* Tempel", are assigned to the non-inflectable word class STOP-WORD of our word model.

The resulting dictionary (GenDicDB) is used to extract all words occurring in the generation output, i.e. all target words specified in the transfer dictionary and the lexical transfer rules. For each part-of-speech a separate dictionary file is generated (g-generation/dic/TDMT_jg.*part-of-speech*).

The entries of the word stems in our example (cf. Chapter 1.3) are listed below.

	stem	*phon*	*stress*	*morph*
verb	("leih"	"l.aI."	"1"	((L-S "leih" (QUOTE (215100 (VGFC C1 . "lieh") (PPRF . "geliehen")))) (L-S "lieh" (QUOTE ((STAMM "leih" . VGFC)))))
	("mo~g"	"m.2:.g."	"1"	((L-S "mo~g" (QUOTE ((WORTART . MODALVERB)(VTYP . 10) (VGFA . A1) (VGFB B3 . "mag") (VGFC C4 . "mochte") (VGFD . "mo~chte") (PPRF . "gemocht") (PAUX . "hab")))) (L-S "mo~chte" (QUOTE ((STAMM "mo~g" . VGFD)))) (L-S "mochte" (QUOTE ((STAMM "mo~g" . VGFC)))) (L-S "mag" (QUOTE ((STAMM "mo~g" . VGFB)))) (L-S "moch" (QUOTE ((STAMM "mo~g" . PPRF))))))
noun	("auto"	"aU.-t.o:."	"10"	((L-S "auto" 130202)))
	("stunde"	"S.t.U.n.-d.@."	"10"	((L-S "stunde" 120103)))
pronoun	("dies"	"d.i:.s."	"1"	((L-S "dies" 1200000)))
	("ich"	"I.C."	"1"	((L-F "ich" (QUOTE (1400000 (FLEXION ((1 ((SG (NOM))))))))))
cardinal	("vier"	"f.i:6.r."	"1"	((L-S "vier" 1900001)))
adverb	("gerne"	"g.E6.-n.@."	"10"	((L-F "gerne" 610000)))
preposition	("fu~r"	"f.y:6.r."	"1"	((L-F "fu~r" 1013000)))

These files contain an entry of the regular stem of each word as well as entries for all irregular stem forms. The size of the current generation dictionary is listed in Table 10.

5.3 Preprocess

Before analyzing the specified target expressions, the syntax of the transfer result is modified in two ways. On the one hand we normalize the given constituent structure for a simpler analysis and on the other hand we shift constituents with a special positioning to the top-level of the respective (sub)sentence. The syntax of the transfer result is modified by applying a set of *adhoc* rules, i.e. simple matching rules which recursively restructure the given input.

During normalization each sentence element with explicitly specified inflection attributes is encapsulated in a ATTR part-of-sentence in order to allow a unique attribute propagation mechanism. Additionally, redundant constituent encapsulation, e.g. in the definition of coordinated or subordinated sentence (SATZREIHE, SATZGEFUEGE), are removed for a simplification of the analysis process.

Due to the lack of examples in our database, the selection of an inappropriate rule during the creation of the target structure can cause main constituents of the target sentence to be encapsulated in other parts of the input, e.g. if a WH phrase, that has to be generated in the pre-field of a constituent questions (WH-Q), is hidden in a noun phrase, it is shifted on top-level in order to handle its positioning in an appropriate way.

Table 10: Size of the Generation Dictionary

TOTAL		8447
verb	VERB	1934
	HILFSVERB	12
	MODALVERB	27
	VERBZUSATZ	323
	PREFIX	323
	SUFFIX	131
noun	NOMEN	5584
	STOP-WORD (proper noun)	2621
adjective	ADJEKTIV	1069
number	KARDINALZAHL	47
article	DETERMINATIV	57
	DETERMINATIV-INDEF	10
pronoun	INTERROGATIVPRONOMEN	13
	PERSONALPRONOMEN	22
	POSSESSIVPRONOMEN	14
	REFLEXIVPRONOMEN	7
	RELATIVPRONOMEN	23
adverb	ADVERB	288
	FRAGEADVERB	38
	PARTIKEL	81
preposition	PRAEPOSITION	119
conjunction	KOORD-KONJUNKTION	13
	SUBORD-KONJUNKTION	68
punctuation	INTERPUNKTION	22

Finally, the sentence type markers specified in the transfer result determines the type of all specified coordinated and subordinated sentences. If no sentence type marker is found, the type SATZ (*statement*) is used as the default type for the generation of the respective sentence.

The syntax of the generation input after the restructuring of the transfer result is listed in Table 11.

5.4 Analysis

After the preprocessing of the transfer result the linguistic information of the specified target expressions are analyzed bottom-up. Given the morphological properties of the basic elements (words and numbers) the part-of-speech and part-of-sentence information are used to analyze the grammatical usage of each constituent (cf. Appendix D).

Words and Numbers

For the classification of a surface word specified in the generation input, we have to extract the stem of the word. If a word contains diacritic characters, they have to be transformed into the internal representation before analyzing the morphological properties of the word stem.

Numbers are specified as digits in the generation input and have to be transformed into their corresponding cardinal word. The classification of cardinal words is carried out on-line in order to restrict the size of the initial generation dictionary[7]. Therefore the digit is arithmetically di-

[7] Numbers defined in the generation dictionary are 0,1,..,19;20,30,...,90;100;1000;1000000.

Table 11: Syntax of the Generation Input

Sen	::=	NIL | MainSen
MainSen	::=	(MainSenTag {MainSenElem}+ {InfAttr}*)
SubordSen	::=	(SubordSenTag {SubordSenElem}+ {InfAttr}*)
CoordClause	::=	(SATZREIHE MainSenTag {MainSenElem}+ {InfAttr}*)
SubordClause	::=	(SATZGEFUEGE SubordSenTag {SenElem}+ {InfAttr}*)
MainSenTag	::=	<cf. Appendix D, Table 25, main clauses>
SubordSenTag	::=	<cf. Appendix D, Table 25, subordinated clauses>
MainSenElem	::=	SenElem | CoordClause
SenElem	::=	SubordClause | PartOfSen | PartOfSp | WordAtom | GenMarker
PartOfSen	::=	({PartOfSenTag | DefPartOfSenTag} {SenElem}+ {InfAttr}*)
PartOfSenTag	::=	<cf. Appendix D, Table 23>
DefPartOfSenTag	::=	*PartOfSenTag
PartOfSp	::=	({PartOfSpTag | DefPartOfSpTag } WordAtom {InfAttr}*)
PartOfSpTag	::=	<cf. Appendix D, Table 22>
DefPartOfSpTag	::=	*PartOfSpTag
GenMarker	::=	<cf. Appendix D, Table 24>
WordAtom	::=	Word | Number
Word	::=	”$String_1$ $String_2$ ...”
String	::=	”...”
Number	::=	1 | 2 | ...
InfAttr	::=	AttrKey AttrValue
AttrKey	::=	<cf. Appendix D, Table 27>
AttrValue	::=	<cf. Appendix D, Table 26>

vided into a sequence of number words (1=”eins”, 2=”zwei”, etc.) and unit words (100=”*hundert*”,1000=”*tausend*”, etc.) separated by a boundary marker[8] ”·”, e.g. ”201” = ”2” $*$ ”100” + ”1” = ”zwei $*$ *hundert*” + ”eins” = ”zwei·*hundert*·eins”).

For each number we have to define the word classification of the cardinal word as well as of the corresponding ordinal word. The ordinal classification is deduced from the cardinal word class. The inflectional characteristics of the ordinal word is equivalent to the ordinal classification of the last number word of the cardinal word. For example, the ordinal stem of the number 1 (”eins”) is ”*erst*”. Thus the ordinal word stem of ”201” (”zwei·hundert·eins”) is ”zwei·hundert·*erst*”, because the last cardinal word of the ”201” (”zwei·hundert·eins”) is ”eins”.

Unknown Words

Words which are not specified in the dictionary are marked as unknown by the prefix character ” ˆ ” of the target word. As listed in Table 1 we distinguish three types according to its character composition. Digits contained in unknown words are generated as cardinal numbers, i.e. the digits are not transformed to their corresponding cardinal words. e.g. ”08-15”. In the case of unit expressions the non-digital parts are generated in lower-case, e.g. ”100m”, whereas unknown words contains no digitals are generated with upper-case characters, e.g. ”SMART”.

[8] The boundary marker is used to reseparate the single cardinal words for the generation of the word phonemes (cf. Chapter 5.7)

Table 12: Analysis of Unknown Words

source marker	target expression	generation
数詞 " ^ アラビア"	KARDINALZAHL "^ ..."	numbers
普通名詞 " ^ アラビア記号"	UNIT "^ ..."	numbers + lower-case
普通名詞 " ^ 頭字語	EIGENNAME "^ ..."	upper-case

Part-of-Speech

The analysis of the part-of-speech tags results in a unique word classification for each word element. If an attribute-value list is specified, the morphological properties of the part-of-speech are restricted to the specified values. If an attribute of the respective word class is not explicitly given, the value defaults to the set of all possible values as defined in Appendix D, Table 27.

Part-of-Sentence

The part-of-sentences summarized in Appendix D, Table 23 combines simple constituents to complex phrases. According to their usage we distinguish the following groups:

grammatical usage → definition of the main linguistic sentence constituents:

(SUB, NOM-OBJ, GEN-OBJ, DAT-OBJ, AKK-OBJ, VP, PP, NP, AP, WH, PLACE, TIME) The subject and object part-of-sentences requires an agreement of its constituents with specific *case* attribute values according to their grammatical function in the sentence. Additionally, in a prepositional phrase (PP, TIME, PLACE) the successor case of the respective preposition has to be propagated to its constituents.

derivative usage → modification of the word classification of the elements in the constituent:

ORDINAL, ORDINAL+, KARDINAL+

transformation of cardinal into corresponding ordinal word and/or numerical extraction

In the case of ORDINAL+ non-cardinal parts are ignored.

Example:
(ORDINAL (KARDINALZAHL "eins") (NOMEN "Tag"))
→ (NP (ORDINALZAHL "erst") (NOMEN "Tag"))
(ORDINAL+ (KARDINALZAHL "eins") (NOMEN "Tag"))
→ (ORDINALZAHL "erst")
(KARDINAL+ (KARDINALZAHL "eins") (NOMEN "Tag"))
→ (KARDINALZAHL "eins")

N+

noun derivation from verbs or adjectives

Priority is given to a derivation form specified in the generation dictionary, but nouns can be derived from verbs by using the infinitive form as stem and assigning neutral gender.

Example:
(N+ (ADJEKTIV "dunkel")) → (NOMEN "Dunkelheit") → GenDic entry
(N+ (VERB "laufen")) → (NOMEN "Lauf") → GenDic entry
(N+ (VERB "laufen")) → (NOMEN "Laufen") → derivation

V+

verb derivation from nouns or adjectives

Example:
(V+ (ADJEKTIV "philosophisch")) → (VERB "philosophieren")
(V+ (NOMEN "Gang")) → (VERB "gehen")

A+, A+pprf, A+pres

adjective derivation from verbs or nouns

A+pres derives the present participle form ("...end"), whereas A+pprf derives the participle perfect form ("ge...").

Example:
(A+ (NOMEN "Pazifik")) → (ADJEKTIV "pazifisch")
(A+pres (VERB "laufen")) → (ADJEKTIV "laufendes")
(A+pprf (VERB "laufen")) → (ADJEKTIV "gelaufenes")

FIX-EXP

definition of idiomatic phrases

no further processing, generated as it is.

Example:
(FIX-EXP "Es tut mir leid .") → "Es tut mir leid."
(FIX-EXP "LaTeX :-) was sonst ?") → "LaTeX :-) was sonst ?"

MAL, MALIG

derivation of cardinal numbers to counting expressions

Example:
(MAL (KARDINALZAHL "eins")) → "einmal"
(MALIG (KARDINALZAHL "zwei")) → "zweimalig"

functional usage → modification of the constituent contents:

SUPERLATIV

superlativ form of adjectives

Example:
(SUPERLATIV (ADJEKTIV "gut")) → "am besten"
(SUPERLATIV (ADJEKTIV "gut") (NOMEN "Tag") → "der beste Tag"

CONCAT

concatenation of successive nouns to a compound noun

The word classification of the last noun is used for the categorization of the new compound word. In analogy to the handling of unknown numbers, a boundary marker "·" is inserted into the concatenation. This boundary is used to reseparate the compound word for the generation of the word phonemes (cf. Chapter 5.7).

No concatenation, if non-nominal parts (besides NOMEN, BUCHSTABE) are included.

Example:
(CONCAT (NOMEN "Hotel") (NOMEN "Zimmer")) → (NOMEN "Hotel·zimmer")
(CONCAT (NOMEN "Bahnhof") (BUCHSTABE "s") (NOMEN "Hotel")) → (NOMEN "Bahnhofs·hotel")

DIGIT

split-off of a specified number into a sequence of simple cardinal words

Example:
(DIGIT "201") → "zwei null eins"
(DIGIT "ˆ 201") → "2 0 1"

AM, PM

generate time expressions by adding an appropriate daytime affix

the expression (AM/PM (KARDINALZAHL "*num*") (UNIT "Uhr")) will be generated as follows:

(AM *num* (UNIT "Uhr"))	$0 \leq num \leq 11$	→ "*num* Uhr morgens"
	$num = 12$	→ "*num* Uhr mittags"
(PM *num* (UNIT "Uhr"))	$num = 0$	→ "12 Uhr mittags"
	$num = 1$	→ "*num* Uhr mittags"
	$2 \leq num \leq 5$	→ "*num* Uhr nachmittags"
	$6 \leq num \leq 12$	→ "*num* Uhr abends"

YEAR

year expression modification (1100 ≤ num ≤ 1999, tenno + year)

In this case the year is expressed by using the century plus the unit word "hundert" plus the decade of the respective year.

Example:
(YEAR (KARDINALZAHL "1901")) → "Neunzehn·hundert·eins"
(YEAR (KARDINALZAHL "2001")) → "Zweitausend·eins"
(YEAR (KARDINALZAHL "12") :tenno :heizei) → "Zweitausend"

ETAGE

convertion of the Japanese floor specifications into European ones

Example:
(ETAGE (KARDINALZAHL "1")) → (NOMEN "Erdgeschoß)"
(ETAGE (KARDINALZAHL "2")) → (NP (ORDINALZAHL "erste") (NOMEN "Etage"))

FLIGHT

flight name vs. number of flights

hints for name specifier are letters, airline names inside phrase, numbers larger then 10.

Example:
(FLIGHT (FIX-UP "KE") (KARDINALZAHL "21")) → "Flug KE21"
(FLIGHT (KARDINALZAHL "2")) → "2 Flüge"

FIX-UP, FIX-CAP

non-inflectionable constituents with capitalization

In the case of FIX-UP all characters are enerated with upper-case characters.

Example:
(FIX-UP "usa") → "USA"
(FIX-CAP "united airlines") → "United Airlines"

NP-END

nominal parts of noun phrase occurring after head noun

suppress insertion of articles in front of noun parts.

Example:
(NP (INITIAL "ICE" (NP-END (NOMEN "Berlin")))) → "ICE Berlin"

DUMMY

suppress the generation of the enclosed constituents

intern usage for *adhoc* restructuring rules (constituent shifted) as well as for the definition of local dictionary in order to suppress redundant meanings in compound noun phrases.

Example:
(NP (NOMEN "Zweibettzimmer") (DUMMY "Zimmer")) → "Zweibettzimmer"

inflectional usage → determination of inflectional attributes:

ATTR

propagate specified attributes to the enclosed constituents

Example:

```
(ATTR (NOMEN "Haus") :number PL)                          → "Häuser"
(SUB (ATTR (NOMEN "Haus") :narticle UNBESTIMMT)) → "ein Haus"
```

EIGENNAME, EIGENNAME+

specification of proper noun phrase

non-German words enclosed are treated as non-inflectable words (STOP-WORD) and inflected in analogy to FIX-CAP part-of-sentences. EIGENNAME+ defines a proper noun phrase without an article insertion.

Example:

```
(EIGENNAME "Kinkakuji Tempel") → (EIGENNAME (FIX-CAP "Kinkakuji")(NOMEN "Tempel"))
(EIGENNAME+ "Japan")              → (ATTR (EIGENNAME "Japan") :narticle OHNE)
```

topological usage → determination of sentence topology

(INTRO, FIX-INTRO, END, FIX-END, AP+, VERB-ADD). This part-of-sentences are used to force a specific position of the encapsulated constituents ignoring the analyzed sentence structure (cf. Table 13).

Subject Analysis

The subject resolution module assignes a subject generation marker (cf. Table 24) to each sentence predicate in the generation input. If multiple markers are assigned to verbal parts we define a priority scheme according to the respective part-of-speech (VERB > MODALVERB > HILFSVERB). However, these marker are only analyzed as the sentence subject, when no other subject (SUB, *SUB) is specified in the generation input. *SUB marker are defined in the transfer knowledge acquisition phase due to errors of the subject resolution module in the context of the training data. However, if an explicit subject (SUB) is specified, all *SUB expressions will be ignored.

Attribute Propagation

After the analysis of the linguistic constituents is completed, we have to adjust the analyzed attribute values of all nouns within each phrase, i.e. we have to determine the scope of all nouns by segmenting each phrase according to the noun positions. This scope is further limited by the location of leading non-inflectional constituents, e.g. conjunctions or prepositions. The noun attributes, which are not explicitly specified in the transfer result, are then propagated within the scope, e.g. the gender of the noun.

Article Insertion

The attribute propagation mechanism is not only used for the analysis of the morphological properties, but also for determing the definiteness of nouns in each phrase, i.e. nouns are associated with an additional attribute *:narticle* whose value determines the article type of these nouns. The analysis of the generation marker on phrase level results in the propagation of the respective *:narticle* attribute. However, the definiteness of a noun can not only be expressed by a definite or an indefinite article. There are some part-of-speeches, e.g. demonstrative pronouns or cardinal numbers, which prevent an article insertion in the scope of the respective noun.

If the definiteness information for a noun can not be resolved, we define a default attribute for the following part-of-sentences, which represents the most common grammatical use of definiteness for these constituents.

definiteness		*part of sentence*
definite	:narticle BESTIMMT	SUB, DAT-OBJ, GEN-OBJ, PP, EIGENNAME
indefinite	:narticle UNBESTIMMT	AKK-OBJ, NOM-OBJ
(force no article)	:narticle OHNE	EIGENNAME+, NP-END

According to the analyzed value of the *:narticle* attribute a corresponding article will be inserted into the leftmost position of the noun scope.

In the case of a definite article insertion the noun phrase is checked for cliticization, i.e. specific prepositions are unified with succeeding definite articles into a contracted preposition, if the article is not stressed. In the current system the following contractions are carried out.

”in” +”das” → ”ins”	”an” +”dem” → ”am”	”von”+”dem” → ”vom”
”um”+”das” → ”ums”	”bei”+”dem” → ”beim”	”zu” +”dem” → ”zum”
”an” +”das” → ”ans”	”in” +”dem” → ”im”	”zu” + ”der” → ”zur”

Update of Topological Fields

After the propagation of the morphological attributes and the insertion of appropriate articles in each phrase, the analysis results of the top-level constituents are updated to the corresponding topological field of the chosen sentence model (cf. Table 13).

Verification of Sentence Structure

The final step of the analysis process is to achieve an agreement between constituents on sentence-level and to check the consistence of the analyzed sentence structure with the chosen sentence model.

Beside the attribute agreement on phrase-level (case attribute for objects, propagation of noun attributes within the noun scope), we also have to propagate the person and number attributes of the subject to the finite verb of the analyzed sentence in order to achieve the required agreement on sentence-level.

Additionally, the attributes of the sentence predicates (tense, number, person, mood, voice) has to agree with the syntactic characteristics of the chosen sentence type (cf. Appendix D, Table 25).

Coordinated and Subordinated Sentences

In the case of a complex sentence, the coordinated and subordinated sentences are generated recursively and the generation result is transformed to a regular constituent of the higher sentence by creating a part-of-speech of type FIX-SEN with the generated subsentence as the pseudo stem. The complete sentence structure of the subsentence is added as the value of the attribute *:structure*.

In contrast to coordinated sentences which are by default updated to the *post field* of the current sentence model, the location of subordinated sentences depends on the structure of the target sentence, e.g. relative clauses are placed directly after the reference noun and thus will be updated to the same topological field as the corresponding noun. However, if the subordinated

Table 13: Correspondence between Topological Fields and Top-Level Constituents

topological field	*top level constituents*
intro	INTRO, FIX-INTRO
F-pre	WH
subject	SUB
nominative object	NOM-OBJ
genitive object	GEN-OBJ
dative object	DAT-OBJ
accusative object	AKK-OBJ
time	TIME
place	PLACE
negation	NOT
reflexive pronoun	REFLEXIVPRONOMEN
F-middle	NP, PP, AP, EIGENNAME, EIGENNAME+, CONCAT FIX-EXP, FIX-CAP, FIX-UP, N+, V+, A+, A+pres, A+pprf AM, PM, YEAR, ORDINAL, ORDINAL+, KARDINAL+, DIGIT ETAGE, FLIGHT, SUPERLATIV, NP-END, DUMMY
finite verb	VP (finite verb)
infinite verb	VP (infinite verb)
verb phrase (adverbial)	AP+
verb phrase (non-verbal)	VP (non verbal parts), VERB-ADD
post field	END, FIX-END

sentence itself forms a top-level constituent, it will be either updated to the *pre field* in order to emphasize the contents of the subordinated sentence, e.g. conditional sentences, or the *post-field* (default).

5.5 Inflection

The analysis of the generation input results in a sentence model, which contains the word classification and inflectional attribute information of each word. Single words are combined to phrases, which are grouped according to the sentence topology.

A single word is inflected by using the word class information coded in the IAL tree of our word model. The IAL tree contains information about the morphological properties of a word suffix depending on the classification of the respective word stem (cf. Figure 6). Given the classification of the word stem and the analyzed inflectional attributes, the IAL tree can be traversed bottom-up in order to determine the corresponding word suffix (cf. Chapter 3.4). The word stem and the selected suffix are concatenated and the retransformation of the internal representation of all diacritics characters results in the generated surface word.

If the analyzed inflectional attributes are underspecified, a default value for the missing attribute is used in order to generate a well-defined surface string (cf. Chapter D, page 77).

The inflection of each topological field is carried out by the concatenation of the generated surface strings of each single word contained in the field.

In the case of the finite verb, the inflection can consist of multiple surface strings, because the respective verb form can be build with the help of auxiliary verbs. Thereby the auxiliary verb is used as the finite verb form and the inflection of the main verb has to be added at the end of the infinite verb field inflection.

According to the topology defined for the chosen sentence type the surface strings of the topological fields are concatenated to the target sentence whereby the punctuation mark defined in the chosen sentence model is added in front and behind the generated sentence string.

Coordinated sentences are by default updated to the *post field* of the higher sentence. Consequently the punctuation mark of the coordinated sentences is succeeded by the punctuation mark of the higher sentence and thus has to be switched in order to achieve the correct assignment.

In contrast to coordinated sentences, the location of subordinated sentences depends on the structure of the target sentence, i.e. the subclause will be updated to either the *pre field* or the *post field* (top-level constituent), or it could be updated to any other topological field, as in the case of relative clauses, which are encapsulated in a top-level constituent. Thus the punctuation marks are required on both sides to clarify the structure of the target sentence for the reader (display of generation result) as well as for the hearer (prosody modeling during synthesis module).

Additionally, an update to a specific topological field can be forced by using a topological part-of-sentence marker, e.g. (INTRO (SATZGEFUEGE ...)) forces the subordinated sentence to be generated in front of the higher sentence.

5.6 Postprocess

The recursive generation of complex sentences leads to an overspecification of punctuation markers in the generated translation string which has to be handled in an appropriate way. Additionally some orthographic modifications, e.g. capitalization of first sentence word and punctuation alignment, have to be carried out for display purposes. The revision of the generation result is done as follows (cf. Appendix D, page 78).

- delete any punctuation marks in front of first sentence word.
- succeeding punctuation marks are replaced by the latest one of this sequence
- insert white space after all remaining punctuation marks (for display only)
- delete word boundary marker ”·” in all words (for display only)
- upcase the first word of each clause in the generation result (for display only)

The postprocess results in an accurately formatted translation string, which forms the output of the generation module to be displayed.

5.7 Interface to Synthesis

The multi-lingual translation approach described in this paper is implemented in a chat translation prototype system which uses the CHATR module for the synthesis of the translation results (cf. [Weeks 97]).

In order to synthesis an adequate speech output various knowledge sources have to be used for the modeling of the intonation and prosody information. The German generation module provides not only the phonemic transcription, syllable boundaries and lexical stress of each word in the generation output, but also the complete morphological word characteristics and the topological structure of the target sentence.

Word Phonemes

The generation dictionary contains the syllabified phonetic information and the lexical stress of each word stem (cf. Chapter 5.2). The phoneme transcription used in the generation dictionary is derived from the *SAMPA* transcription of the CELEX lexical database and mapped to the *sampaG* phone-set of the German speakers defined in the CHATR system which contains the following phones.

”6”	”@”	”2:”	”2:6”	”9”	”96”	”a”	”a6”	”a:”	”a:6”	”aI”
”aU”	”E”	”E6”	”E:”	”E:6”	”e:”	”e:6”	”I”	”I6”	”i:”	”i:6”
”O”	”O6”	”OY”	”OY6”	”o:”	”o:6”	”U”	”U6”	”u:”	”u:6”	”Y”
”Y6”	”y:”	”y:6”	”a ”	”O ”	”E ”	”9 ”	”C”	”N”	”S”	”b”
”d”	”f”	”g”	”h”	”j”	”k”	”l”	”m”	”n”	”p”	”r”
”s”	”t”	”v”	”x”	”z”	”Z”					

A word stem phoneme is a sequence of phones separated by the phoneme boundary marker ”.”, which are grouped to syllables separated by the syllable boundary marker ”–”.

The lexical stress patterns consist of a sequence of ”0” and ”1”. Each number corresponds to one syllable of the word phoneme, whereby a syllable is stressed, if it is marked with ”1”.

In analogy to the generation of the surface strings (concatenation of word stem and appropriate suffix), the word phoneme is created using the word stem phoneme and the phonetic transcription of the generated suffix[9]. If the last syllable of the word-stem and the first syllable of the suffix phoneme are combined to one single syllable, the lexical stress information of the surface word has to be merged in an appropriate way. Another modification of the surface phoneme is necessary, if the final phoneme is a voiced consonant (”b”,”d”,”g”,”v”,”z”). In this case the consonant has to be replaced by its corresponding voiceless counterpart as listed below.

”b.” → ”p.”	”v.” → ”f.”	”I.g.” → ”I.C.”
”d.” → ”t.”	”z.” → ”s.”	”r.” → ””
”k.” → ”g.”	”Z.” → ”S.”	

Additionally, the phoneme ending ”I.g.” of a surface word ending in ”g” has to be replaced by ”I.C.” and a final ”r.” consonant has to be deleted in order to achieve an accurate pronounciation of the word.

In the case of compound nouns, created on-line (number, CONCAT, YEAR), the word boundary marker ”·” is used to reseparate the simple words in order to create the word phoneme of the compound noun as a sequence of the word phonems of the simple words.

G-FILTER Interface

The format of the interface between TDMT and CHATR is the same for all generation modules of TDMT. It consists of a string, whose contents is divided into several lines. Each line contains four fields (separated by an ”|” character) representing the orthographic, syntactic, phonetic and stress information of the generated word or phrase. However, there are differences between the module interfaces concerning the contents of the fields and their use within CHATR. If a information cannot be provided by the generation output, the respective field is empty.

The German generation module carries out a detailed morphological and syntactic analysis of the target sentence. Thus we can provide not only words, but also the respective phoneme and stress information as well as the complete morphological and structural information of the target constituents.

[9] All possible suffixes are defined in *g-generation/dic/tdmt_jg.SUFFIX*

field 1 contains the surface word information, i.e. a list of words separated by white spaces.

field 2 contains the linguistic information of the word phrase of field 1. This field is divided into three subfields, separated by a ":" character. Subfield 1 contains the type of the sentence in which the phrase occurs. Subfield 2 specifies the topological field of the chosen sentence model. Subfield 3 consists of a sequence of part-of-speech tags which characterize the word classification of the corresponding surface words listed in field 1.

field 3 contains the syllabified phonetic information, i.e. a list of syllabified phonemes separated by white spaces. Each single word phoneme corresponds to the words of field 1.

field 4 contains the stress pattern information, i.e. a list of stress patterns separated by white spaces. Each single stress pattern corresponds to the words of field 1.

Punctuations mark the boundary of a (sub)sentence and are used to predict the prosody of the given sentence. Thus they are listed as a single line entry and used to insert pauses into the synthesis output.

Below, the German input to the G-FILTER interface is listed for the example sentence "この車を四時間借りたいんですができますでしょうか" (cf. Chapter 1.3).

```
    words         type    topology                   part of speeches                 phonemes                 stress
"
ich               |SATZ :SUBJECT        :PERSONALPRONOMEN                   |I.C.                           |1
moechte           |SATZ :V-FIN          :MODALVERB                          |m.9.C.-t.@.                    |10
gerne             |SATZ :VP-ADVERB      :ADVERB                             |g.E6.-n.@.                     |10
dieses auto       |SATZ :AKK-OBJECT     :DETERMINATIV NOMEN                 |d.i:.-s.@.s. aU.-t.o:.         |10 10
fuer vier stunden|SATZ :F-MIDDLE        :PRAEPOSITION KARDINALZAHL NOMEN|f.y:6. f.i:6. S.t.U.n.-d.@.n.|1 1 10
leihen            |SATZ :V-INF          :VERB                               |l.aI.-@.n.                     |10
.                 |SATZ :BOUNDARY       :INTERPUNKTION                      |4                              |0
ist               |YN-Q :V-FIN          :HILFSVERB                          |I.s.t.                         |1
es                |YN-Q :SUBJECT        :PERSONALPRONOMEN                   |@.s.                           |0
moeglich          |YN-Q :VP-NON-VERB:ADJEKTIV                               |m.2:.k.-l.I.C.                 |10
?                 |YN-Q :BOUNDARY       :INTERPUNKTION                      |4                              |0
"
```

How the knowledge contained in the G-FILTER interface is used to create real speech is decribed in detail in [Brinckmann 97] and [Striegnitz 97].

6 Evaluation

Currently, the TDMT system addresses dialogues in the *travel domain*, such as travel scheduling, hotel reservations, and trouble-shooting. We have applied TDMT to four language pairs: Japanese-English, Japanese-Korean [Furuse et al. 96], Japanese-German [Paul et al. 98] and Japanese-Chinese [Yamamoto 99]. Table 14 shows the transfer knowledge statistics.[10] Training and test utterances were randomly selected per dialogue from our speech and language data collection that includes about 40 thousand utterances in the travel domain [Takezawa 99]. The coverage of our training data differs among the language pairs and varies between about 3.5% and about 9%.

[10] The development of the KJ system is suspended. The JC system has just been started so it is still too early to evaluate it. Other directions, CJ and GJ have not yet been implemented.

Table 14: Transfer Knowledge Statistics

Count	JE	JG	JK	EJ
Words	15063	15063	15063	7937
Patterns	1002	802	801	1571
Examples	16725	9912	9752	11401
Trained Utterances	3639	1917	1419	3467

Table 15: Quality and Time

	JE	JG	JK	EJ
A (%)	43.4	45.8	71.0	52.1
A+B (%)	74.0	65.9	92.7	88.1
A+B+C (%)	85.0	86.4	98.0	95.3
Time (Seconds)	0.09	0.13	0.05	0.05

6.1 The Evaluation Procedure

A system dealing with spoken-dialogues is required to realize a quick and informative response that supports smooth communication. Even if the response is somewhat broken, there is no chance for manual pre/post-editing of input/output utterances. In other words, both speed and informativity are vital to a spoken-language translation system. Thus, we evaluated TDMT's translation results for both *time* and *quality.*

Three native speakers of each target language manually graded translations for 23 *unseen* dialogues (330 Japanese utterances and 344 English utterances, each about 10 words). During the evaluation, the native speakers were given information not only about the utterance itself but also about the previous context. The use of context in an evaluation, which is different from typical translation evaluations, is adopted because the users of the spoken-dialogue system consider a situation naturally in real conversation.

Each utterance was assigned one of four ranks for translation quality: (A) Perfect: no problems in both information and grammar; (B) Fair: easy-to-understand with some unimportant information missing or flawed grammar; (C) Acceptable: broken but understandable with effort; (D) Nonsense: important information has been translated incorrectly. Here we show samples for each rank containing information about 1. input, 2. system translation, 3. human translation, and 4. explanation.

6.2 Results

Table 15 shows the latest evaluation results for TDMT, where the "acceptability ratio" is the sum of the (A), (B) and (C) ranks. The JE and JG translations achieved about 85% acceptability and the JK and EJ translations achieved about 95% acceptability. JK's superiority is due to the linguistic similarity between the two languages; EJ's superiority is due to the relatively loose grammatical restrictions of Japanese.

The translation speed was measured on a PC/AT PentiumII/450MHz with 1GB of memory. The translation time did not include the time needed for a morphological analysis, which is much faster

Table 16: Transfer Knowledge Statistics for JG

Count	96/7	97/2	97/9	98/2	99/2
Words	4937	10048	12790	12790	15063
Patterns	623	787	815	766	802
Examples	2935	5039	6975	7824	9912
Trained Utterances	931*	1553*	1928*	2053*	1917

Table 17: Quality Improvement of JG

	96/7	97/2	97/9	98/2	99/2
A (%)	17.2	26.7	33.3	39.0	45.8
A+B (%)	31.0	46.0	48.9	56.1	65.9
A+B+C (%)	48.4	64.6	68.9	75.5	86.4

than a translation. Although the speed depends on the amount of knowledge and the utterance length, the average translation times were around 0.1 seconds.

6.3 Progress of JG

During 4 years of development we conducted 5 evaluations as described above. In Table 16 the increase of transfer knowledge is listed. In 1998 we adjusted the JG system to the transfer knowledge definition (lexical and local transformation as well as patterns) of the JE system resulting in a drop of the amount of patterns in the statistics for the 98/2 evaluation.

Moreover, besides the last evaluation the number of training data is given based on sentence unit (*). However, in 1998 the training database was restructured according to utterance units, i.e. the training data "はい分かりました。それで結構です。" was counted as 2 training sentences in previous evaluations, whereas only 1 utterance was counted in the last evaluation. Thus the training utterance number 1917 of the 99/2 evaluation does still represent an increase of the training data amount.

The progress of the JG development is documented in Table 17. The translations of rank A are still less than 50% and mainly consists of short sentences. As soon as the complexity/length of the utterances increases the translation performance decreases. Even if the JG performance for rank A and A+B+C sentences is similar to the JE result, we got 10% less performance for rank A+B. Thus, even if we can get an "acceptable" translation, it might not be a natural, easy-to-understand one. This is mainly due to a lack of examples extracted from the training data which frequently causes a selection of semantically far examples resulting in a poorer translation quality. Furthermore, there is still a deficit concerning the coverage of pattern expressions occurring in every-days Japanese. Thus, we expect an increase of the translations performance according to an increase of the training amount due to the improvement of their naturalness.

6.4 Pending Problems

Besides the lack of training data, there are quite a lot of problems which can't be dealed with in the current system and have to be addressed in future research in order to improve the translation performance. Pending problems concerning the transfer knowledge are described in detail in [Paul & Yamazaki 99].

Word Level

number of noun phrases

An open problem is the handling of number features of Japanese noun phrases which do not explicitly express this feature. Moreover, numerical modifiers in Japanese are frequently separated from the noun they modify.

definiteness

The insertion of articles in TDMT is based on default assumptions concerning part-of-speech tags as well as grammatical markers. However, instead of such ad-hoc solutions a more linguistically based analysis for the definite, indefinite or generic usage of nouns is required in the future.

elision

The generation dictionary is automatically derived from the CELEX database which contains elision information (cf. section 3.1.1). However, the deletion of a vowel in the inflection form might sound unnatural for the evaluator (depending on his/her linguistic background/origin), thus has a negative influence to the "acceptance" of the translation. Therefore, these entries have to be checked manually.

Phrase Level

phrase structure

For various generation aspects (attribute propagation, article insertion, etc.) as well as repair strategies it is necessary to know the structure of complex phrases, i.e. we have to incorporate phrase structure boundaries into our current approach. So far, a flat structure can only be obtained by analyzing grammatical part-of-sentence markers as well as defining compound noun phrases according to the CONCAT maker within the current generation framework.

transitiveness

In the current system the transitiveness of verbs are marked by adding default objects, e.g. *AKK-OBJ, to the transfer result. These are either defined in the transfer dictionary or inserted during the transfer knowledge application. However, dealing with transitiveness requires the analysis of verb case structure as well as reflexive/infinitive verb information.

case assignment

The constituents of the generation input are analyzed separately according to their linguistic markers. So far, we are not able to make any kind of consistency check in order to check the grammatical relation between verbal parts and verb case constituents, e.g. in the case of PP attachment we have to check the correctness of successor preposition or in the case of the main verb *sein* ("to be") the object has to be marked as a NOM-OBJ case.

Sentence Level

subordination of clauses

Most of the early training data consist of short sentences which didn't cause to much trouble to the syntactic definition of the German sentence models. However, complex sentences (frequently occurring in the CstarII training data set) require a more elaborated handling of intra-sentential relationships between co-/subordinated sentences, e.g. the analysis of non-specified subject in subordinated clauses.

modality

At the moment we don't have any knowledge about the speaker's intention about his/her utterances. Therefore, we don't whether the provided is a fact the speaker knows about, a wish, a belief etc. Consequently we are not able to deal with modality which leads to an inadequate translation of modality expressions.

Context Processing

speaker information

Recently, information about who is speaking (male vs. female) and the speakers role (clerk vs. customer) is utilized in the EJ translation module of TDMT (cf. [Yamada et al. 00]). Similar to EJ, the information about the speakers gender is also used, for example, for the translation of 様, さん expressions in JE (cf. [Tokuhashi 99]). However, this kind of extra-linguistic knowledge still has to be incorporated into the JG transfer knowledge.

coreferential relationships

The translations of the current system are restricted to unique utterances only. However, information about references within the current dialog would enable us not only to adjust the attributes of anaphoric expressions, e.g. gender in German, but would also result in a context-adopted word selection for other sentence constituents, e.g. verbs.

dialog structure

Information about the dialog structure, e.g. what is the type of dialog act of a specific utterances (statement, request, ...), could be used to achieve adequate translation of even short utterances like "いいえ。".

Verification

The accuracy of the topological field approach used in the German generation module depends on the well-formedness of the target structure analyzed in the transfer module. So far, the generation module generates the linguistic constituents specified in the transfer result without checking syntactic and semantic constraints in the context of the target sentence, e.g. there is no check of obligatory and optional complements required in the given verbal context. Thus, future work has to focus on the prevention, detection and recover of structural errors due to the lack of appropriate examples in our training data.

References

[Brinckmann 97] C. Brinckmann. 1997. German in Eight Weeks - A Crash Course for CHATR. ATR Technical Report TR-IT-0236, ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan.

[Busemann 83] S. Busemann. 1983. Oberflächentransformationen bei der automatischen Generierung geschriebener deutscher Sprache. Master's thesis, Fachbereich Informatik, Univ. Hamburg.

[Finkler & Neumann 88] W. Finkler and G. Neumann. 1988. MORPHIX: A Fast Realization of a Classification-Based Approach to Morphology. In H. Trost, editor, *4. Österreichische Artificial-Intelligence-Tagung: Wiener Workshop - Wissensbasierte Sprachverarbeitung*, p. 11–19. Springer, Berlin, Heidelberg.

[Furuse et al. 96] O. Furuse, J. Kawai, H. Iida, S. Akamine, and D. Kim. 1996. Multi-lingual Spoken-Language Translation Utilizing Translation Examples. In *Proc. of NLPRS'95*, p. 544–549, Seoul, Korea.

[Furuse & Iida 96] O. Furuse and H. Iida. 1996. Incremental Translation Utilizing Constituent Boundary Patterns. In *Proc. of the 16th COLING*, p. 412–417, Copenhagen, Denmark.

[Furuse & Masui 95] O. Furuse and A. Masui. 1995. 言語データベース概要. ATR Technical Report TR-IT-0136, ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan.

[Helbig & Buscha 72] G. Helbig and J. Buscha. 1972. *Deutsche Grammatik.* Verlag Enzyklopädie, Leipzig.

[Ono & Hamanishi 81] S. Ono and M. Hamanishi, editors. 1981. 角川類語新辞典. 角川書店.

[Paul et al. 98] M. Paul, E. Sumita, and H. Iida. 1998. Field Structure and Generation in Transfer-Driven Machine-Translation. In *Proc. of 4th Annual Meeting of the NLP*, p. 504 – 507, Fukuoka, Japan.

[Paul & Yamamoto 00] M. Paul and K. Yamamoto. 2000. Resolution of Referential Expressions within TDMT. ATR Technical Report TR-IT-0327, ATR Interpreting Telecommunications Laboratories, Kyoto, Japan.

[Paul & Yamazaki 99] M. Paul and Y. Yamazaki. 1999. TDMT 解析変換知識作成の手引 (日独版) - Instructions for building TDMT's transfer knowledge (Japanese → German). ATR Technical Report TR-IT-0310, ATR Interpreting Telecommunications Laboratories, Kyoto, Japan.

[Piepenbrock 95] R. Piepenbrock, 1995. *CELEX Lexical database (Dutch, English,German), Version 2.5.* Max Planck Institute of Psycholinguistics, Nijmegen, Netherlands.

[Quinlan 93] J. Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

[Schott 72] G. Schott. 1972. Automatic Analysis of Inflectional Morphemes in German Nouns. *Acta Informatica*, 1:360–374.

[Singer 97] H. Singer. 1997. Atrsprec. http://www.itl.atr.co.jp/~singer/software/SPRECDOC/release.html.

[Striegnitz 97] K. Striegnitz. 1997. Teaching CHATR German Intonation - Lesson One. ATR Technical Report TR-IT-0237, ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan.

[Sumita et al. 99] E. Sumita, S. Yamada, K. Yamamoto, M. Paul, H. Kashioka, K. Ishikawa, and S. Shirai. 1999. Solutions to Problems Inherent in Spoken-language Translation: The ATR-MATRIX Approach. In *Proc. of the Machine Translation Summit VII*, p. (to appear), Singapore.

[Sumita & Iida 92] E. Sumita and H. Iida. 1992. Example-based Transfer of Japanese Adnominal Particles into English. *IEICE Trans*, E75-D, No.4:585–594.

[Takezawa 99] T. Takezawa. 1999. Building a bilingual travel conversation database for speech translation research. In *Proc. of Oriental COCOSDA Workshop'99.*

[Tokuhashi 99] N. Tokuhashi. 1999. TDMT 解析変換知識作成の手引 (日英版). Manual, ATR Interpreting Telecommunications Laboratories, Kyoto, Japan.

[Weeks 97] M. Weeks. 1997. Chatr - a generic speech synthesis system. http://www.itl.atr.co.jp/~mweeks/.

[Yamada et al. 00] S. Yamada, E. Sumita, and H. Kashioka. 2000. 話者の役割を利用した音声翻訳. In *Proc. of 6th Annual Meeting of the NLP*, p. (to appear), Kanazawa, Japan.

[Yamamoto 99] K. Yamamoto. 1999. Proofreading Generated Outputs: Automated Rule Acqusition and Application to Japanese-Chinese Machine Translation. In *Proc. of the 18th ICCPOL*, p. 87–92, Tokushima, Japan.

A Morphology

Table 18: Japanese Part-of-Speeches

Part-of-Speech		description
本動詞	ほんどうし	verb (分かる)
助動詞	じょどうし	auxiliary verb (ます, たい)
準体助動詞	じゅんたいじょどうし	complementary auxiliary verb (でしょう, みたい)
補助動詞	ほじょどうし	complementary verb (ございます, になる)
判定詞	はんていし	copula verb (です, じゃ)
普通名詞	ふつうめいし	common noun (部屋, フィリップス)
サ変名詞	さへんめいし	common noun; predicative use with する (予約)
形容名詞	けいようめいし	adjectival common noun (必要, 大丈夫)
サ変形容名詞	さへんけいようめいし	adjectival common noun; predicative use with する (失礼, 心配)
形式名詞	けいしきめいし	noun without any content; always occurring with modifying expressions (ところ, こと)
形容詞	けいようし	adjective (詳しい, 面白い)
連体形容詞	れんたいけいようし	adjectival (+ ”な) adjective (大きな)
連体詞	れんたいし	adjectival determiner (その, そんな)
数詞	すうし	cardinal number (四, 零, 〇, 万)
代名詞	だいめいし	pronoun (こちら, あなた, どれ)
副詞	ふくし	adverb (そう, どうも, 少々)
ローマ字	ろうまじ	letter (エー, ピー)
係助詞	かかりじょし	topic particle (は, も)
格助詞	かくじょし	case particle (が, を, に)
終助詞	しゅうじょし	sentence ending particle (ね, が)
準体助詞	じゅんたいじょし	nominalization particle (高いの)
副助詞	ふくじょし	adverbial particle (とか, など)
連体助詞	れんたいじょし	compound particle (と言う)
接続助詞	せつぞくじょし	conjunction particle (けれど, ので)
並立助詞	へいりつじょし	noun conjunction particle (と, や)
感動詞	かんどうし	interjection (はい, うん, 恐れ入ります)
記号	きごう	punctuation (。)
準数詞	じゅんすうし	expression substituting a cardinal (十何個)
人称接尾辞	にんしょうせつびじ	address form suffix (様, たち)
接続詞	せつぞくし	conjunction (では, そして)
接続副詞	せつぞくふくし	conjunction adverb (また, なお)
接頭辞	せっとうじ	prefix (お, 御, 約)
接尾辞	せつびじ	suffix (さ, 的)

B Transfer

Table 19: Bigram Markers

</kihon-adj>	</kihon-adjnoun>	</kihon-cn>	</kihon-fn>	</kihon-fn-auxv>
</kihon-fukujo>	</kihon-pron>	</kihon-rentai>	</kihon-sn>	</kihon-v>
</kihon-yotei-auxv>	<adj/kihon-cn>	<adj/kihon-fn>	<adj/kihon-fn-auxv>	<adj/kihon-sn>
<adj/kihon-v>	<adj/renyou-adj>	<adj/renyou-adjnoun>	<adj/renyou-adv>	<adj/renyou-cn>
<adj/renyou-exp>	<adj/renyou-sn>	<adj/renyou-v>	<adjnoun->	<adjnoun-conj>
<adv->	<adv-adj>	<adv-adjnoun>	<adv-adv>	<adv-cn>
<adv-pron>	<adv-sn>	<adv-v>	<auxv->	<auxv/renyou->
<auxv/renyou-adv>	<auxv/renyou-cn>	<auxv/renyou-interj>	<auxv/renyou-v>	<cn-adj>
<cn-adjnoun>	<cn-adv>	<cn-cn>	<cn-conj>	<cn-conjadv>
<cn-exp>	<cn-num>	<cn-prenom>	<cn-pron>	<cn-rentai>
<cn-rentaiadj>	<cn-roman>	<cn-sadjnoun>	<cn-sn>	<cn-v>
<conjadv->	<conjpp->	<end>	<endp->	<exp->
<fn-adj>	<fn-adv>	<fn-cn>	<fn-exp>	<fn-pron>
<fn-sn>	<fn-v>	<haku-nichi>	<hi-you>	<juntaip-adv>
<na-adj>	<na-cn>	<na-fn>	<na-fn-auxv>	<na-sn>
<no-yotei-pred>	<num-cn>	<num-num>	<num-pron>	<num-youbi>
<personpnom-adv>	<personpnom-cn>	<personpnom-exp>	<personpnom-rentai>	<personpnom-sn>
<personpnom-v>	<pnom-cn>	<pnom-conj>	<pnom-interj>	<pnom-pron>
<pnom-roman>	<pnom-sn>	<pnom-v>	<pron-adjnoun>	<pron-adv>
<pron-cn>	<pron-conj>	<pron-exp>	<pron-pron>	<pron-rentai>
<pron-sn>	<pron-v>	<rentaiadj-adjnoun>	<rentaiadj-cn>	<roman->
<roman-roman>	<sadjnoun->	<sadjnoun-conj>	<sn-adjnoun>	<sn-adv>
<sn-cn>	<sn-conj>	<sn-exp>	<sn-fn>	<sn-num>
<sn-pron>	<sn-sn>	<sn-v>	<sn-yotei-pred>	<sou/auxv>
<sou/jauxv>	<subp->	<subp-patch>	<tsuki-hi>	<v-imp>
<v/renyou-adjnoun>	<v/renyou-cn>	<v/renyou-exp>	<v/renyou-fn>	<v/renyou-hojov>
<v/renyou-sn>	<v/renyou-v>	<>		

Table 20: Rule Categories

p – predicate	i – interjection	s – sentence
n – noun	d – determiner	e – modal (end)
a – adverb	terminal – compound	m – modal

category-set	part-of-speeches/categories
word-n	普通名詞 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
word-p	本動詞 形容詞 補助動詞 感動詞 連体形容詞
word-all	サ変名詞 形容名詞 サ変形容名詞 形式名詞 普通名詞 数詞 準数詞 代名詞 本動詞 補助動詞 形容詞 連体形容詞 副詞 接続副詞 連体詞 接続詞 感動詞 助動詞 準体助動詞 判定詞 格助詞 準体助詞 係助詞 副助詞 並立助詞 接続助詞 終助詞 連体助詞 接頭辞 接尾辞 人称接尾辞 記号 ローマ字 その他
group-n	s+n n+n dn nd n 普通名詞 代名詞 サ変名詞 サ変形容名詞 形容名詞 形式名詞 数詞 準数詞 ローマ字 副詞 接続副詞
group-p	is ss np sp ap pm nm s+m p 本動詞 形容詞 感動詞 サ変名詞 形容名詞 サ変形容名詞 連体形容詞
group-s	is ss np se sm sp ap pm nm s+m p 本動詞 形容詞 感動詞

category	part-of-speeches/categories
terminal	word-n word-p terminal
p	本動詞 形容詞 形容名詞 普通名詞
pm	pm nm s+m p 本動詞 形容詞 サ変名詞 形容名詞 サ変形容名詞 副詞 接続副詞
n	n word-n 副詞 接続副詞
nd	n+n nd n word-n 副詞 接続副詞
dn	s+n n+n dn nd n word-n 形容詞 連体形容詞 副詞 接続副詞
n+n	s+n n+n dn nd n word-n 副詞 接続副詞
np	np sp ap pm nm s+m p word-p group-n 副詞 接続副詞
ap	np sp ap pm nm s+m p word-p group-n 副詞 接続副詞
nm	副詞 接続副詞 group-n
s+n	np ap sp pm nm s+m p s+n n+n dn nd n word-n word-p 副詞 接続副詞
s+m	np ap sp pm nm s+m p word-p 副詞 接続副詞
sp	ss sp np ap sm pm nm s+m p word-p group-n 副詞 接続副詞
sm	sm 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
se	sm 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
ss	is ss np se sm np sp ap pm nm s+m p word-p
is	is ss np se sm sp np ap pm nm s+m s+n n+n dn nd n p サ変名詞 形容名詞 サ変形容名詞 形式名詞 普通名詞 数詞 準数詞 代名詞 本動詞 補助動詞 形容詞 連体形容詞 副詞 接続副詞 連体詞 接続詞 感動詞 助動詞 準体助動詞 判定詞 格助詞 準体助詞 係助詞 副助詞 並立助詞 接続助詞 終助詞 連体助詞 接頭辞 接尾辞 人称接尾辞 記号 ローマ字 その他

Table 21: Rule Patterns

("あと" "で" ?x)
("いろんな" ?x)
("こういう" ?x)
("ご" ?x)
("すると" ?x)
("そういたしますと" ?x)
("そうしますと" ?x)
("そして" ?x)
("それ" "は" ?x)
("それだったら" ?x)
("それと" ?x)
("ただし" ?x)
("できれ" "ば" ?x)
("では" ?x)
("ところで" ?x)
("どの" ?x "も" ?y)
("のぞみ" ?x "号")
("丸" ?x "日")
("午後" ?x "時")
("御" ?x)
("大変" <> ?x "ことは多い")

("あと" <> ?x)
("お" ?x)
("この" ?x)
("しかし" ?x)
("そういうことでしたら" ?x)
("そういった" ?x)
("そうすると" ?x)
("そしてまた" ?x)
("それから" ?x)
("それで" ?x)
("それなら" ?x)
("ちょっとした" ?x)
("でしたら" ?x)
("でも" ?x)
("どういう" ?x)
("なかなか" <adjnoun-> ?x "ない")
("ひかり" ?x "号")
("顔色" "が" ?x)
("午後" ?x "時" ?y "分")
("合わ" "せ" "て" ?x)
("申し訳ありません" "が" ?x)

("あの" ?x)
("お" ?x "する")
("このような" ?x)
("じゃ" ?x)
("そういう" ?x)
("そうしたら" ?x)
("そこで" ?x)
("その" ?x)
("それが" ?x)
("それでしたら" ?x)
("それに" ?x)
("で" ?x)
("ですから" ?x)
("といいますのも" ?x)
("どういった" ?x)
("どのような" ?x)
("よろしけれ" "ば" ?x)
("気分" "が" ?x)
("午前" ?x "時")
("時間" "が" ?x)
("実は" ?x)

("あれ" <pron-cn> ?x)
("お客様" <> ?x)
("こんな" ?x)
("すみません" "が" ?x)
("そういたしましたら" ?x)
("そうしましたら" ?x)
("そしたら" ?x)
("そのような" ?x)
("それじゃ" ?x)
("それでは" ?x)
("そんな" ?x)
("でき" "たら" ?x)
("ですが" ?x)
("ということは" ?x)
("どの" ?x)
("どんな" ?x)
("一刻" "も" ?x)
("恐れ入ります" "が" ?x)
("午前" ?x "時" ?y "分")
("時間" "を" ?x)
("第" ?x)

```
("第" ?x "種")
("約" ?x)
(?x "終わる")
(?x "う")
(?x "か" "の" ?y)
(?x "か" <subp-> ?y)
(?x "かしら")
(?x "かどうか" <endp-> ?y)
(?x "かね")
(?x "かもしれない")
(?x "から" "の" <no-yotei-pred> ?y)
(?x "から" ?y "に" ?z)
(?x "からは" ?y)
(?x "が" ?y)
(?x "が" ?y <cn-adj> ?z)
(?x "がる")
(?x "けども")
(?x "ことができる")
(?x "ことになる")
(?x "ことは多い")
(?x "さん")
(?x "しか" <> ?y "は" "ございません")
(?x "しか" <subp-> ?y "ていません")
(?x "ずに" <> ?y)
(?x "せていただける")
(?x "せてもらえる")
(?x "そう" <sou/auxv>)
(?x "た")
(?x "た" </kihon-fn-auxv> ?y)
(?x "た" </kihon-rentai> ?y)
(?x "たい")
(?x "たことはある")
(?x "たばかり")
(?x "たら" "良い")
(?x "だ")
(?x "だけ")
(?x "だった")
(?x "っけ")
(?x "て")
(?x "て" "よ")
(?x "て" "欲しい")
(?x "てある")
(?x "ていた")
(?x "ていただける")
(?x "ていました")
(?x "ています" "の")
(?x "ていません" "か")
(?x "ているのです")
(?x "ておりました")
(?x "ております" "の")
(?x "ておる")
(?x "てから" ?y)
(?x "てくる")
(?x "てございます")
(?x "てしまう")
(?x "てまいる")
(?x "てみる")
(?x "ても" ?y)
(?x "てもらえる")
(?x "で" "良い")
(?x "で" ?y "を" ?z)
(?x "である")
(?x "できる")
(?x "でございましょう")
(?x "でございます" "か")
(?x "でし" "たら" ?y)
(?x "でした" "か")
```

```
("第" ?x <num-youbi> ?y)
(?X の)
(?x "あり" "ません")
(?x "か")
(?x "か" "も" ?y)
(?x "か" <subp-patch> ?y)
(?x "かと" ?y)
(?x "かな")
(?x "かねる")
(?x "かもしれません")
(?x "から" "の" ?y)
(?x "から" ?y "の" "間")
(?x "からも" ?y)
(?x "が" ?y "が" ?Z)
(?x "が" ?y <cn-v> ?z)
(?x "くらい")
(?x "けれども")
(?x "ことにする")
(?x "ことはある")
(?x "ごと")
(?x "し" ?y)
(?x "しか" <> ?y "ない")
(?x "すぎる")
(?x "せいたす")
(?x "せてください")
(?x "せてもらっている")
(?x "そう" <sou/jauxv> "です")
(?x "た" "よう")
(?x "た" </kihon-fn> "ところ")
(?x "た" <> "方" "が" ?y)
(?x "たいと思う")
(?x "たことはない")
(?x "たほうがよい")
(?x "たら" ?y)
(?x "だ" "と" ?y)
(?x "だけ" "は" ?y "という" ?z)
(?x "だら" "良い")
(?x "って")
(?x "て" "は" ?y)
(?x "て" "よろしい")
(?x "て" ?y)
(?x "ていく")
(?x "ていただきたい")
(?x "ていただける" "ば" ?y)
(?x "ています")
(?x "ています" <> ?y)
(?x "ていらっしゃる")
(?x "ているのです" "か")
(?x "ております")
(?x "ております" <> ?y)
(?x "てかえる")
(?x "てください")
(?x "てくれる")
(?x "てございません")
(?x "ての" ?y)
(?x "てまわる")
(?x "ても" "良い")
(?x "てもらいたい")
(?x "で")
(?x "で" ?y)
(?x "で" ?y <> ?z)
(?x "でいらっしゃる")
(?x "でございまし" "た")
(?x "でございましょう" "か")
(?x "でございます" "と" ?y)
(?x "でしか" ?y "ない")
(?x "でしたね")
```

```
("歩い" "て" ?x)
(?X ます)
(?x "いたす")
(?x "か" "で" ?y)
(?x "か" <endp-> ?y)
(?x "かい")
(?x "かどうか" "も" ?y)
(?x "かな" "と" ?y)
(?x "かは" ?y)
(?x "から")
(?x "から" ?y)
(?x "からに" ?y)
(?x "が")
(?x "が" ?y <adjnoun-> ?z)
(?x "が" ?y <pron-v> ?z)
(?x "けど")
(?x "けれども" ?y)
(?x "ことになっている")
(?x "ことはない")
(?x "さ")
(?x "しか" <> "無い")
(?x "しか" <> ?y "ません")
(?x "ずつ")
(?x "せていただく")
(?x "せてもらう")
(?x "せる")
(?x "そして" ?y)
(?x "た" </kihon-cn> ?y)
(?x "た" </kihon-fn> ?y)
(?x "た" <> "方<ホウ>" "が" ?y)
(?x "たいと思っている")
(?x "たところだった")
(?x "たら" "と" ?y)
(?x "たり")
(?x "だい")
(?x "だっ" "たら" ?y)
(?x "だら" ?y)
(?x "っぽい")
(?x "て" "ほしい" "と" ?y)
(?x "て" "結構")
(?x "て" ?y "たほうがよい")
(?x "ていける")
(?x "ていただく")
(?x "ていない")
(?x "ています" "か")
(?x "ていません")
(?x "ている")
(?x "ておく")
(?x "ております" "か")
(?x "ておりません")
(?x "てから" "で" ?y)
(?x "てくださる")
(?x "てこれる")
(?x "てさしあげる")
(?x "てはいけない")
(?x "てまわれる")
(?x "ても" "良い" "なら" ?y)
(?x "てもらう")
(?x "で" "いう" "と" ?y)
(?x "で" ?y "で" ?z)
(?x "であり" "ます" </kihon-sn> ?y)
(?x "でき" "たら" "と" ?y)
(?x "でございまし" "たら" ?y)
(?x "でございます")
(?x "でございますね")
(?x "でした")
(?x "でしょう")
```

```
(?x "でしょう" "か")
(?x "です" "か")
(?x "です" "けれども" ?y)
(?x "ですね")
(?x "ではありません" "か")
(?x "ではなく" ?y)
(?x "でも" <subp-> ?y)
(?x "と" "あと" <> ?y)
(?x "と" "それから" ?y)
(?x "と" "一緒" "に" ?y)
(?x "と" "言う" "ます" "と" ?y)
(?x "と" ?y "では" ?z)
(?x "という" "こと")
(?x "という" "名前" "の" ?y)
(?x "というの" "は" ?y)
(?x "とか")
(?x "としたら" ?y)
(?x "となる")
(?x "な")
(?x "ない" "かと" ?y)
(?x "なおす")
(?x "なくてはいけない")
(?x "なくなる")
(?x "など" "の" ?y)
(?x "なのです")
(?x "なのです" "けれども" ?y)
(?x "ならではの" ?y)
(?x "なんて")
(?x "に" "し" "て" ?y)
(?x "に" "乗っ" "て" ?y)
(?x "に" ?y "は" ?z)
(?x "にこしたことはない")
(?x "についても" ?y)
(?x "にでも" ?y)
(?x "になれる")
(?x "によって" "も" ?y)
(?x "に比べて" ?y)
(?x "ねがえる")
(?x "の" "こと")
(?x "の" "でしょう" "か")
(?x "の" "ほう")
(?x "の" "を" ?y)
(?x "の" "他" "に" ?y)
(?x "の" "方<カタ>")
(?x "の" "無い" "よう" "に" ?y)
(?x "の" ?y "から" ?z)
(?x "の" ?y "と" ?z)
(?x "のか" <endp-> ?y)
(?x "ので" ?y)
(?x "のです" "か")
(?x "のみ")
(?x "は")
(?x "は" ?y "が" ?z)
(?x "は" ?y "と" ?z)
(?x "は" ?y "ので" ?z)
(?x "は" ?y <cn-pron> ?z)
(?x "ば")
(?x "ば" ?y)
(?x "へ" ?y)
(?x "へは" ?y)
(?x "ほどではない")
(?x "まし" "て" ?y)
(?x "ましょ" "う" "か")
(?x "ます" "の")
(?x "ますね")
(?x "ません" "か")
(?x "まで" ?y)
(?x "でしょうね")
(?x "です" "から" ?y)
(?x "です" "と" ?y)
(?x "での" ?y)
(?x "ではない")
(?x "でも")
(?x "でも" ?y)
(?x "と" "する")
(?x "と" "ついでに" ?y)
(?x "と" "言う" "と" ?y)
(?x "と" "良い")
(?x "と" ?y "の" ?z)
(?x "という" "ふう" "に" ?y)
(?x "という" ?y)
(?x "といった" "の")
(?x "とか" <> "あと" <> ?y)
(?x "として" ?y)
(?x "とは" ?y)
(?x "な" "こと")
(?x "ない" "よう" <> ?y)
(?x "ながら")
(?x "なくてはいけません")
(?x "なさる")
(?x "なのでしょう")
(?x "なのです" "か")
(?x "なのですね")
(?x "なんか")
(?x "に")
(?x "に" "基づい" "て" ?y)
(?x "に" ?y)
(?x "において" "は" ?y)
(?x "にしか" ?y "ていません")
(?x "につき" ?y)
(?x "になる")
(?x "には" ?y)
(?x "によって" ?y)
(?x "に面した" ?y)
(?x "の")
(?x "の" "だ")
(?x "の" "ではない" "かな")
(?x "の" "よう")
(?x "の" "次" "に" ?y)
(?x "の" "方")
(?x "の" "方<ホウ>")
(?x "の" <no-yotei-pred> ?y)
(?x "の" ?y "が" ?z)
(?x "の" ?y <cn-cn> "前")
(?x "のそばの" ?y)
(?x "のでし" "た")
(?x "のですね")
(?x "のような" ?y)
(?x "は" "ない")
(?x "は" ?y "しか" <> ?z)
(?x "は" ?y "に" ?z)
(?x "は" ?y "は" ?z)
(?x "は" ?y <cn-v> ?z)
(?x "ば" "よろしい")
(?x "ばかり")
(?x "へと" ?y)
(?x "べき")
(?x "まし" "た")
(?x "ましょ" "う")
(?x "ます")
(?x "ます" "よ")
(?x "ません")
(?x "まで")
(?x "までに" ?y)
(?x "です")
(?x "です" "が" ?y)
(?x "です" "とか" <> "あるいは" "また" <> ?y)
(?x "ではありません")
(?x "ではない" "かと" ?y)
(?x "でも" "よろしい")
(?x "でも" <subp-> ?y "でも")
(?x "と" "し" "まし" "て" "は" ?y)
(?x "と" "ですね" <auxv-> ?y)
(?x "と" "言う" "ます" "か" <> ?y)
(?x "と" ?y)
(?x "と" ?y "は" ?z)
(?x "という" "わけ" "ではない")
(?x "というの")
(?x "といった" ?y)
(?x "とか" <subp-> ?y)
(?x "としては" ?y)
(?x "と別に" ?y)
(?x "ない")
(?x "ないでください")
(?x "ながら" ?y)
(?x "なくてはいけません" "か")
(?x "など")
(?x "なのでしょう" "か")
(?x "なのです" "が" ?y)
(?x "なら" ?y)
(?x "なんか" "の" ?y)
(?x "なんて" <subp-> "いう" </kihon-fn> ?y)
(?x "に" "合う" "せる" "て" ?y)
(?x "に" ?y "が" ?z)
(?x "にくい")
(?x "について" ?y)
(?x "にて" ?y)
(?x "になる" "ます" "と" ?y)
(?x "にも" ?y)
(?x "による" ?y)
(?x "ね")
(?x "の" "かわり" "に" ?y)
(?x "の" "でしょう")
(?x "の" "ところ" "に" ?y)
(?x "の" "よう" "に" ?y)
(?x "の" "所" "に" ?y)
(?x "の" "方" "が" ?y)
(?x "の" "方<ホウ>" "が" ?y)
(?x "の" ?y)
(?x "の" ?y "でも" <subp-> ?z)
(?x "のある" ?y)
(?x "ので")
(?x "のです")
(?x "のに" ?y)
(?x "の中の" ?y)
(?x "は" ?y)
(?x "は" ?y "で" ?z)
(?x "は" ?y "の" ?z)
(?x "は" ?y "を" ?z)
(?x "は" ?y <pron-v> ?z)
(?x "ば" "良い")
(?x "ばと" ?y)
(?x "への" ?y)
(?x "ほど")
(?x "まし" "た" "か")
(?x "まし" "た" </kihon-fn> "ところ")
(?x "ます" "か")
(?x "ます" </kihon-cn> ?y)
(?x "ます" </kihon-rentai> ?y)
(?x "まで" "の" ?y)
(?x "みたい")
```

```
(?x "みたい" "です")
(?x "も" ?y)
(?x "も" ?y <adjnoun-> ?z)
(?x "もうしあげる")
(?x "やすく" <auxv/renyou-> ?y)
(?x "よう" "でし" "たら" ?y)
(?x "よう" <> ?y)
(?x "より" ?y)
(?x "らしさ")
(?x "を" "使っ" "て" ?y)
(?x "を" ?y)
(?x "を" ?y <cn-exp> ?z)
(?x "ん" "が" ?y)
(?x "ドル")
(?x "マイル")
(?x "ヶ月間")
(?x "駅" "目")
(?x "階")
(?x "軒")
(?x "個")
(?x "才")
(?x "時" "半")
(?x "時間" "半")
(?x "種類")
(?x "食")
(?x "席")
(?x "達")
(?x "的")
(?x "等")
(?x "日" "の" ?y)
(?x "年")
(?x "泊" <haku-nichi> ?y "日")
(?x "番街")
(?x "部屋")
(?x "便")
(?x "名")
(?x "枡")
(?x </kihon-cn> "中" "で" ?y)
(?x </kihon-fn> ?y)
(?x </kihon-sn> ?y)
(?x <> "あと" "は" ?y)
(?x <> "そして" ?y)
(?x <> "なし")
(?x <> "以下")
(?x <> "以上" "の" ?y)
(?x <> "共")
(?x <> "頃")
(?x <> "前" "に" ?y)
(?x <> "対" <> ?y)
(?x <> "発")
(?x <> "分")
(?x <> "方" "の" ?y)
(?x <> "方<ホウ>" "の" ?y)
(?x <> ?y "が" ?z)
(?x <adj/kihon-cn> ?y)
(?x <adj/kihon-fn> ?y)
(?x <adj/renyou-adjnoun> ?y)
(?x <adj/renyou-exp> ?y)
(?x "みたいな" ?y)
(?x "も" ?y "は" ?z)
(?x "も" ?y <cn-pron> ?z)
(?x "や" ?y)
(?x "よ")
(?x "よう" "に" ?y)
(?x "ような" ?y)
(?x "よりも" ?y)
(?x "れる")
(?x "を" "出" "てから" "の" ?y)
(?x "を" ?y "が" ?z)
(?x "を" ?y <cn-v> ?z)
(?x "ウォン")
(?x "パーセント")
(?x "マルク")
(?x "ヶ所")
(?x "円")
(?x "基")
(?x "軒" "先")
(?x "号")
(?x "冊")
(?x "時" ?y "分")
(?x "時間" ?y "分")
(?x "周")
(?x "人")
(?x "台")
(?x "中")
(?x "度")
(?x "等" "席")
(?x "日" <hi-you> ?y)
(?x "泊")
(?x "晩")
(?x "番線")
(?x "分")
(?x "本")
(?x "様")
(?x </kihon-adj> ?y)
(?x </kihon-cn> ?y)
(?x </kihon-pron> ?y)
(?x </kihon-v> ?y)
(?x <> "および" ?y)
(?x <> "それに" ?y)
(?x <> "または" ?y)
(?x <> "以外")
(?x <> "以内")
(?x <> "行き")
(?x <> "乗り")
(?x <> "前" "までに" ?y)
(?x <> "中")
(?x <> "発" <> ?y <> "行き")
(?x <> "方" "が" ?y)
(?x <> "方<ホウ>" "が" ?y)
(?x <> "未満")
(?x <> ?y <> "当たり" "で" ?z)
(?x <adj/kihon-fn-auxv> ?y)
(?x <adj/kihon-sn> ?y)
(?x <adj/renyou-adv> ?y)
(?x <adj/renyou-sn> ?y)
(?x "め")
(?x "も" ?y "も" ?z)
(?x "も" ?y <cn-v> ?z)
(?x "やすい")
(?x "よう")
(?x "よう" "になる")
(?x "よね")
(?x "ら")
(?x "わ")
(?x "を" "利用し" "て" ?y)
(?x "を" ?y <cn-cn> ?z)
(?x "を" ?y <subp-> ?z)
(?x "セント")
(?x "ブロック")
(?x "ヶ月")
(?x "駅")
(?x "回")
(?x "系統")
(?x "軒" "目")
(?x "号室")
(?x "時")
(?x "時間")
(?x "社")
(?x "週間")
(?x "世紀")
(?x "大" ?y)
(?x "丁目")
(?x "度" ?y "分")
(?x "日")
(?x "日間")
(?x "泊" "する")
(?x "番")
(?x "部")
(?x "分間")
(?x "枚")
(?x "列" "目")
(?x </kihon-adjnoun> ?y)
(?x </kihon-fn-auxv> ?y)
(?x </kihon-rentai> ?y)
(?x </kihon-yotei-auxv> ?y)
(?x <> "する")
(?x <> "ところ" <> ?y)
(?x <> "もの" "で" ?y)
(?x <> "以上")
(?x <> "気味")
(?x <> "込み")
(?x <> "前")
(?x <> "前後")
(?x <> "程度")
(?x <> "付")
(?x <> "方" "で" ?y)
(?x <> "方<ホウ>" "で" ?y)
(?x <> "目")
(?x <adj/kihon-cn> "ほう" "が" ?y)
(?x <adj/kihon-fn> "ところ")
(?x <adj/kihon-v> ?y)
(?x <adj/renyou-cn> ?y)
(?x <adj/renyou-v> "ございます")
```

```
(?x <adj/renyou-v> "ございます" "か")
(?x <adj/renyou-v> "なれ" "ば" "なる" "ほど" <subp-> ?y)
(?x <adjnoun-> ?y)
(?x <adv-> ?y)
(?x <adv-adjnoun> ?y)
(?x <adv-cn> ?y)
(?x <adv-sn> ?y)
(?x <auxv-> ?y)
(?x <cn-adj> ?y)
(?x <cn-adv> ?y)
(?x <cn-conj> ?y)
(?x <cn-conjadv> ?y)
(?x <cn-num> ?y)
(?x <cn-pron> ?y "か" "で" ?z)
(?x <cn-pron> ?y <pron-v> ?z)
(?x <cn-rentaiadj> ?y)
(?x <cn-sadjnoun> ?y)
(?x <cn-sn> ?y <sn-exp> ?z)
(?x <conjadv-> ?y)
(?x <end>)
(?x <exp-> ?y)
(?x <fn-cn> ?y)
(?x <fn-pron> ?y)
(?x <fn-v> ?y)
(?x <na-adj> ?y)
(?x <na-fn-auxv> ?y)
(?x <na-sn> ?y)
(?x <num-num> ?y)
(?x <personpnom-adv> ?y)
(?x <personpnom-exp> ?y)
(?x <personpnom-v> ?y)
(?x <pnom-sn> ?y)
(?x <pron-adv> ?y)
(?x <pron-cn> ?y "が" ?z)
(?x <pron-rentai> ?y)
(?x <pron-v> ?y)
(?x <rentaiadj-cn> ?y)
(?x <roman-roman> ?y)
(?x <sn-adjnoun> ?y)
(?x <sn-exp> "おねがいします")
(?x <sn-fn> ?y)
(?x <sn-pron> ?y)
(?x <sn-v> ?y)
(?x <subp-> ?y)
(?x <v-imp>)
(?x <v/renyou-exp> ?y)
(?x <v/renyou-sn> ?y)
(?x <adj/renyou-v> ?y)
(?x <adj/renyou-v> ?y "ない")
(?x <adjnoun-> ?y <cn-v> ?z)
(?x <adv-adj> ?y)
(?x <adv-adv> ?y)
(?x <adv-pron> ?y)
(?x <adv-v> ?y)
(?x <auxv/renyou-> ?y)
(?x <cn-adjnoun> ?y)
(?x <cn-cn> ?y)
(?x <cn-conj> ?y "を" ?z)
(?x <cn-exp> ?y)
(?x <cn-pron> ?y)
(?x <cn-pron> ?y "を" ?z)
(?x <cn-rentai> ?y)
(?x <cn-roman> ?y)
(?x <cn-sn> ?y)
(?x <cn-v> ?y)
(?x <conjpp-> ?y)
(?x <endp-> ?y)
(?x <fn-adv> ?y)
(?x <fn-exp> ?y)
(?x <fn-sn> ?y)
(?x <juntaip-adv> ?y)
(?x <na-cn> ?y)
(?x <na-fn> ?y)
(?x <num-cn> ?y)
(?x <num-pron> ?y)
(?x <personpnom-cn> ?y)
(?x <personpnom-sn> ?y)
(?x <pnom-cn> ?y)
(?x <pnom-v> ?y)
(?x <pron-cn> ?y)
(?x <pron-exp> ?y)
(?x <pron-sn> ?y)
(?x <rentaiadj-adjnoun> ?y)
(?x <roman-> ?y)
(?x <sadjnoun-> ?y)
(?x <sn-cn> ?y)
(?x <sn-exp> ?y)
(?x <sn-num> ?y)
(?x <sn-sn> ?y)
(?x <sn-yotei-pred> ?y)
(?x <tsuki-hi> ?y "日")
(?x <v/renyou-adjnoun> ?y)
(?x <v/renyou-fn> "次第")
```

C German Word Analysis/Inflection

C.1 Lexem Definition

Noun

(l-s *Stem* **abcdef**)
(l-s *Stem* ’(**abcd00** (PLURAL . *PluralStem*)))
(l-s *PluralStem* ’(**abcd01** (STAMM *Stem* . PLURAL)))
(l-s *Stem* ’({ NounClassification }+))

a	b	c	d	ef
1	1 = MAS 2 = FEM 3 = NTR	0 = no 1 = yes	1 ... 9 (cf. Table 2a)	01 ... 13 (cf. Table 2b)
part of speech	gender	umlaut	singular class	plural class

Verb

(l-s *Stem* ’(**abcdef** { VerbClassification }+))

a	bc	d	ef
2	00 (prefix + basis verb) 01 ... 95 (12*VGFA)+verb-type	0 = ”sein” + no prefix 1 = ”haben” + no prefix 2 = ”sein” + prefix 3 = ”haben” + prefix	number
part of speech	VGFA class + verb-type (cf. Table 4 and 5)	aux-verb + prefix	length of prefix

Remark: verb + prefix + no split off → **d** = 0/1, **ef** = length of prefix

Adjective

(l-s *Stem* **abcdef**)

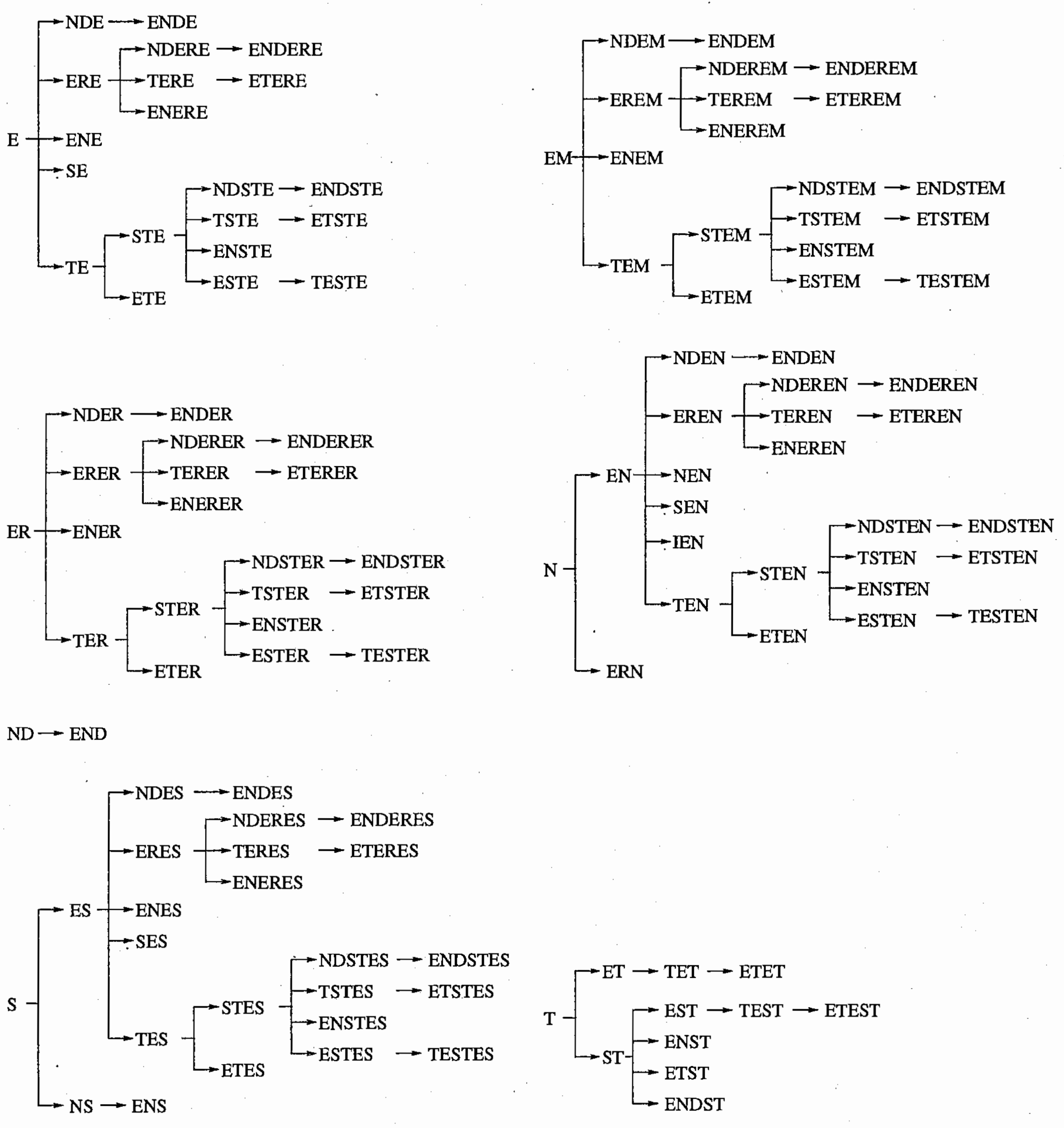
a	b	c	d	e	f
3	0 = no KOM,SUP 1 = ”st” 2 = ”est” 3 = irregular KOM, SUP 4 = irregular POS,KOM,SUP	0 = no 1 = necessary 2 = possible	0 = no 1 = necessary 2 = possible	0 = no 1 = yes	1 = A1 2 = A2 3 = A3 4 = B1 5 = B2 6 = B3 7 = C1
part of speech	comparison (cf. Table 6b)	umlaut (cf. Table 6c)	elision (cf. Table 6d)	stem-end ”e” predicative	adj. class (cf. Table 6a)

fullform-lexicon entries

(l-f *Stem* **ab0000**)

a	6 = **Adverb**	7 = **Particle**	8 = **Coord-Conj**	9 = **Subord-Conj**
b	0 = no additional category exists		1 = additional category exists	

C.2 Suffix Tree



C.3 Top-level Functions

morphix–read *input-string* function

Use: Read input and transform it to the internal representation, used within the MORPHIX sub-module, i.e. down-case of the input-string and encoding of special characters (e.g. the German umlaut).

Arguments: *input-string* a string of a (possibly) multiple words, divided by spaces.

Return: A list of single word-strings, transformed to the internal representation.

Remarks: The special characters are transformed as follows:

```
ä, Ä → a˜    ß→ s˜
ö, Ö → o˜    é → e'
ü, Ü → u˜    ê → e^    ...
```

The German "ß" is not transformed, when it occurs at the end of the word and there are inflection forms, where "ss" is used instead of "ß" (e.g. "Faß", but "Fässer" in the plural form).

Example:

```
(morphix–read "übergroß")  ⇒  "u˜bergros˜"
(morphix–read "Café")      ⇒  " cafe'"
(morphix–read "Faß")       ⇒  "fass"
```

word–analysis *word* function

Use: Morphological analysis of the specified word.

Arguments: *word* a string, that has to be analyzed.

Return: The internal representation of all possible morphological analysis results of the input word.

Remarks: Function call abbreviation: *(w–a word)*

Example:

```
                  (word–analysis "sein")
                            ⇓
(("sei" (WORTART HILFSVERB) (FLEXION ((INFINITIV))))
 ("sein" (WORTART POSSESSIVPRONOMEN)
       (FLEXION ((ARTIKELWORT ((MAS ((SG (NOM)))
                                (NTR ((SG (NOM AKK))))
                               ))))))

                (word–analysis "ha˜usern")
                            ⇓
(("haus" (WORTART NOMEN) (FLEXION ((NTR ((PL (DAT)))))))
```

dialog-for *word* function

Use: In the case that a input cannot be analyzed correctly, a clarification dialog will be started in order to "learn" the correct inflection forms of the input (cf. Chapter 3.2). Several question will be asked about the word category, etc. and the given information will be saved to the user-lexicon and used for upcoming morphological analysis.

Arguments: *word* a string, whose morphological information has to be learned.

Return: The internal representation of the learned word analysis or NIL, if the clarification dialog is aborted.

Remarks: The clarification dialog can be aborted at any time (input="U"). The clarification dialog can be disabled by setting the global variable ***clarification-dialog-on***=NIL.

Function call abbreviation: *(c–d word)*

---

clarification-dialog-on global variable

Use: Disable (value=NIL) or enable (value=T) the invoke of an clarification dialog, if an input cannot be analyzed.

Remarks: The disabling is for example necessary during the translation process of the TDMT system, because the standard-output stream will be suppressed during the translation process. When the clarification dialog is disabled and an input cannot not be analyzed, an error message will be given and the return value is NIL.

---

userlex global variable

Use: Pathname of the user-defined lexicon. All, through clarification dialog learned lexical-items, will be saved to this lexicon.

Remarks: Default directory defined in: (make-system-path "g-generation;dic;add")

C.4 Inflection Functions

---

adjective–inflection *stem comparation attributive–used–p*
&optional *article gender number case* function

Use: generation of the positive and comparison forms of attributive and predicative used adjectives. The values of the arguments are restricted to the following ones.

Arguments:

stem	string, representing a canonical adjective-stem
comparation	POS KOM SUP
attributive–used–p	NIL T
article	OHNE BESTIMMT UNBESTIMMT
gender	MAS FEM NTR
number	SG PL
case	NOM GEN DAT AKK

Return: A string; the inflection of the adjective, given the specified attributes.

Example: (adjective–inflection "alt" SUP T BESTIMMT MAS SG DAT)

⇓

"ältesten"

detdef–inflection *case number gender* &optional *det* function

Use: generation of the definite articles (det=DET) and relative pronouns (det=T). The values of the arguments are restricted to the following ones.

Arguments:

case	NOM GEN DAT AKK
number	SG PL
gender	MAS FEM NTR
det	DET T

Return: A string; the inflection of the definite article or relative pronoun, given the specified attributes.

Example: (detdef–inflection GEN SG MAS DET)

⇓

”des”

(detdef–inflection GEN SG MAS T)

⇓

”dessen”

determiner–inflection *stem case number gender* function

Use: generation of all possible determiner. The values of the arguments are restricted to the following ones.

Arguments:

stem	string, representing a canonical determiner-stem
case	NOM GEN DAT AKK
number	SG PL
gender	MAS FEM NTR

Return: A string; the inflection of the determiner, given the specified attributes.

Example: (determiner–inflection ”jen” NOM SG NTR)

⇓

”jenes”

determiner–indef–inflection *stem case number gender* function

Use: generation of all possible indefinite determiner. The values of the arguments are restricted to the following ones.

Arguments:

stem	string, representing a canonical indefinite determiner-stem
case	NOM GEN DAT AKK
number	SG PL
gender	MAS FEM NTR

Return: A string; the inflection of the indefinite determiner, given the specified attributes.

Example: (determiner–indef–inflection "irgendein" GEN SG NTR)

⇓

"irgendeines"

---

noun–inflection *stem case number* — function

Use: generation of all possible conjugation-forms of nouns. The values of the arguments are restricted to the following ones.

Arguments:

stem	string, representing a canonical noun-stem
case	NOM GEN DAT AKK
number	SG PL

Return: A string; the inflection of the noun, given the specified attributes.

Example: (noun–inflection "haus" NOM PL)

⇓

"Häuser"

---

ordinal–inflection *stem attributive–used–p* &optional *article gender number case* — function

Use: generation of the ordinal numbers. The values of the arguments are restricted to the following ones.

Arguments:

stem	string, representing a canonical ordinal-stem
attributive–used–p	NIL T
article	OHNE BESTIMMT UNBESTIMMT
gender	MAS FEM NTR
number	SG PL
case	NOM GEN DAT AKK

Return: A string; the inflection of the ordinal, given the specified attributes.

Example: (ordinal–inflection "zwei" T OHNE MAS SG NOM)

⇓

"zweiter"

---

query–inflection *case number gender* — function

Use: generation of the WH expression (e.g. "wer", "was"). The values of the arguments are restricted to the following ones.

Arguments:

case	NOM GEN DAT AKK
number	SG
gender	MAS FEM NTR

Return: A string; the inflection of the WH expression, given the specified attributes.

Example: (query–inflection NOM SG NTR)
⇓
”was”

(query–inflection DAT SG MAS)
⇓
”wem”

perspron–inflection *person case number* &optional *gender* function

Use: generation of the personal pronouns. The values of the arguments are restricted to the following ones.

Arguments:

person	1 2 2a 3
case	NOM GEN DAT AKK
number	SG PL
gender	MAS FEM NTR

Return: A string; the inflection of the personal pronoun, given the specified attributes.

Example: (perspron–inflection 3 DAT SG FEM)
⇓
”ihr”

(perspron–inflection 2a NOM SG)
⇓
”Sie”

possessive–inflection *stem gender number case* &optional *type article* function

Use: generation of the possessive pronouns, which can be either used as an article word, i.e. in front of a noun instead of an article, or as a noun itself. The values of the arguments are restricted to the following ones.

Arguments:

stem	string, representing a canonical possessive-pronoun-stem
gender	MAS FEM NTR
number	SG PL
case	NOM GEN DAT AKK
type	ARTIKELWORT SUBSTANTIVWORT
article	OHNE BESTIMMT

Return: A string; the inflection of the possessive pronoun, given the specified attributes.

Example: (possessive–inflection ”mein” MAS SG NOM SUBSTANTIVWORT OHNE)
⇓
”meiner”

(possessive–inflection ”mein” MAS SG NOM ARTIKELWORT)
⇓
”mein”

reflexive–inflection *person case number* function

Use: generation of the reflexive pronouns. The values of the arguments are restricted to the following ones.

Arguments:

person	1 2 2a 3
case	DAT AKK
number	SG PL

Return: A string; the inflection of the reflexive pronoun, given the specified attributes.

Example: (reflexive–inflection 1 DAT SG)

⇓

”mir”

verb–inflection *stem tense mood voice person number* function

Use: generation of all possible conjugation-forms of verbs, modal-verbs and auxiliary-verbs. The values of the arguments are restricted to the following ones.

Arguments:

stem	string, representing a canonical verb-stem
tense	PRAESENS IMPERFEKT PERFEKT FUTUR–1 FUTUR–2 PLUSQUAMPERFEKT
mood	INDIKATIV KONJUNKTIV IMPERATIV
voice	AKTIV PASSIV
person	1 2 2a 3
number	SG PL

Return: The inflection of the verb, given the specified attributes. In the case of compound tenses, the returned value is a list of inflected strings. Otherwise a single string will be returned.

Example: (verb–inflection ”geh” PRAESENS INDIKATIV AKTIV 2 SG)

⇓

”gehst”

(verb–inflection ”hingeh” PRAESENS INDIKATIV AKTIV 2 SG)

⇓

(”gehst” ”hin”)

(verb–inflection ”seh” PERFEKT KONJUNKTIV PASSIV 2 PL)

⇓

(”seiet” ”gesehen” ”worden”)

C.5 Visualization Functions

show-adjective *stem* function

Use: List the inflection forms for the characteristics: comparison (POS, KOM, SUP), article (OHNE, BESTIMMT, UNBESTIMMT), case (NOM, GEN, DAT, AKK), person (MAS, FEM, NTR) and number (SG,PL).

Arguments: *stem* a string, representing a canonical adjective stem.

Remarks: Function call abbreviation: *(s–a stem)*.

---

show-imperativ *stem* function

Use: List the imperative inflection forms for the characteristics: number (SG,PL), address form (ANREDE) and voice (AKTIV,PASSIV).

Arguments: *stem* a string, representing a canonical verb stem.

Remarks: Function call abbreviation: *(s–i stem)*.

---

show-noun *stem* function

Use: List the inflection forms for the characteristics: case (NOM, GEN, DAT, AKK) and number (SG, PL).

Arguments: *stem* a string, representing a canonical noun stem.

Remarks: Function call abbreviation: *(s–n stem)*.

---

show-possessives *stem* function

Use: List the inflection forms for the characteristics: gender (MAS, FEM, NTR), case (NOM, GEN, DAT, AKK) and number (SG, PL).

Arguments: *stem* a string, representing a canonical possessive-pronoun stem.

Remarks: Function call abbreviation: *(s–p stem)*.

---

show-persprons *person* function

Use: List the inflection forms for the characteristics: case (NOM, GEN, DAT, AKK) and number (SG, PL).

Arguments: *person* a symbol, representing a valid person specifier.

Remarks: Function call abbreviation: *(s–pe person)*.

---

show-verb *stem* function

Use: List the inflection forms of all tenses of the specified stem.

Arguments: *stem* a string, representing a canonical verb stem.

Remarks: Function call abbreviation: *(s–v stem)*.

D Generation

Table 22: German Part-of-Speeches

category	part-of-speech (品詞)	description
verb	VERB	verb (本動詞)
	HILFSVERB	auxiliary verb (助動詞)
	MODALVERB	modal verb (話法の助動詞)
	VERBZUSATZ	verb prefix (動詞の前綴)
noun	NOMEN	common noun (普通名詞)
	INITIAL	initial (かしら文字)
	BUCHSTABE	letter (文字)
	UNIT	unit (単位)
adjective	ADJEKTIV	adjective (形容詞)
	KARDINALZAHL	cardinal number (基数)
	ORDINALZAHL	ordinal number (序数)
pronoun	POSSESSIVPRONOMEN	possessive pronoun (所有代名詞)
	PERSONALPRONOMEN	personal pronoun (人称代名詞)
	REFLEXIVPRONOMEN	reflexive pronoun (再帰代名詞)
	DETERMINATIV	determiner (<指示>代名詞)
	DETERMINATIV-INDEF	indefinite determiner (不定代名詞)
	RELATIVPRONOMEN	relative pronoun (関係代名詞)
	INTERROGATIVPRONOMEN	interrogative pronoun (疑問代名詞)
adverb	ADVERB	adverb (副詞)
	FRAGEADVERB	interrogative adverb (疑問副詞)
particle	PARTIKEL	particle (不変化詞)
preposition	PRAEPOSITION	preposition (前置詞)
conjunction	KOORD-KONJUNKTION	coord. conjunction (並列接続詞)
	SUBORD-KONJUNKTION	subord. conjunction (従属接続詞)
others	INTERPUNKTION	punctuation (記号)
	STOP-WORD	unknown word (未知語)

Table 23: German Part-of-Sentences

usage	part-of-sentence	description
grammatical	SUB	subject (force nominative case)
	NOM-OBJ	nominative object (force nominative case)
	GEN-OBJ	genitive object (force genitive case)
	DAT-OBJ	dative object (force dative case)
	AKK-OBJ	accusative object (force accusative case)

grammatical	VP	verb phrase
	NP	noun phrase
	PP	prepositional phrase (force successor case of preposition)
	AP	adverbial phrase
	WH	interrogative phrase
	PLACE	locative expression
	TIME	temporal expression
derivative	ORDINAL	transform enclosed cardinal number into an ordinal one
	ORDINAL+	extract/transform enclosed cardinal number to an ordinal one
	KARDINAL+	extract enclosed cardinal number
	N+	nominalization of verbs/adjectives
	V+	verbalization of nouns/adjectives
	A+/A+pprf/A+pres	adjectivalization of nouns/verbs (pprf/pres participle)
	FIX-EXP	fixed expression, no analysis on word-level
	MAL	adverb derivation from cardinal with stem form ”...mal”
	MALIG	adjective derivation from cardinal with stem form ”...malig”
functional	SUPERLATIV	superlative form (”am X-ten” vs. ”X-te Nomen)”
	CONCAT	concatenate nouns to a compound noun using boundary ”·” (”Hotel” (*hotel*),”Zimmer” (*room*) ⇒ ”Hotel·zimmer”(*hotel room*))
	DIGIT	defining enclosed cardinal number as a chain of single numbers
	AM	transform time expression into ”a.m.”, i.e. add daytime affix
	PM	transform time expression into ”p.m.”, i.e. add daytime affix
	YEAR	transform a number (1100≤n≤2000) into its year-expression (”1997” = ”19” ∗ ”100” + ”97” ⇒ ”19”·”Hundert”·”97”)
	ETAGE	transform Japanese floor expression to the German one
	FLIGHT	specification for a noun phrase concerning flight information (”Flug X” vs. ”X Flüge”)
	FIX-CAP	capitalized fixed expression, no analysis on word-level
	FIX-UP	upper-cased fixed expression, no analysis on word-level
	NP-END	additional phrase after noun without any article insertion
	DUMMY	suppress output of word (due to redundancy in compound NP)
inflectional	ATTR	specifying inflection attributes for enclosed parts
	EIGENNAME	proper noun phrase; unknown words are treated as FIX-CAP
	EIGENNAME+	proper noun phrase without any article insertion
topological	INTRO	leading expression, used for conjunction, interjection, etc.
	FIX-INTRO	leading expression, no analysis on word-level
	END	ending expression; used for heavy NP-shift, subord.sentences
	FIX-END	ending expression, no analysis on word-level
	AP+	adverbial phrase with special positioning
	VERB-ADD	verb phrase addition (equivalent to non-verbal parts of VP)

Table 24: Generation Markers

category	generation marker	description
negation	{NOT}	sentence negation
sentence type	{COORD}	sentence coordination
article	{OHNE} {BESTIMMT} {UNBESTIMMT}	no insertion of an article insert definite article insert indefinite article

pronoun	{POSS}
	{MEIN} / {IHR}
		possessive pronoun (subject agreement)
		1.person / address form

address form	{HERR} {FRAU} {VORNAME}	male addressee female addressee first name addressee

subject
(automatically inserted by HOKAN module)
	{<AGEN S>}
	{<AGEN K>}
	{<AGEN H>}
	{<AGEN Y>}
	{<AGEN G>}
	{<AGEN X>}
	{<AGEN F>}
		1.person, singular (”ich”)
		1.person, plural (”wir”)
		2.person, address form / singular (”Sie”/”du”)
		2.person, address form / plural (”Sie”/”ihr”)
		3.person, general person (”man”)
		3.person, singular (”er/sie/es”)
		3.person, singular, neutral (”es”)

Table 25: German Sentence Types

main clauses					
type	finite verb	tense	mood	person	end
SATZREIHE	—	coordination of main clauses			
SATZGEFUEGE	—	coordination of at least one subordinated clause			
SATZ	:second	—	INDIKATIV KONJUNKTIV	—	”.”
SATZ_verb-first	:first	statement preceded by subordinated sentence			”.”
FRAGE	—	WH part-of-sentence is included → yes (WH-Q), no (YN-Q)			
WH-Q	:second	—	INDIKATIV KONJUNKTIV	—	”?”
YN-Q	: first	—	INDIKATIV KONJUNKTIV	—	”?”
FRAGE_verb-first	:first	question preceded by subordinated sentence			”?”
IMP-S-I	:first	PRAESENS	IMPERATIV	2 2a	”.”
IMP-S-II	:first	PRAESENS	IMPERATIV	2 2a	”!”
IMP-S_verb-first	:first	imperative preceded by subordinated sentence			”!”
DES-S-I	:second	—	KONJUNKTIV	3	”.”
DES-S-II	:first	—	KONJUNKTIV	3	”!”

subordinated clauses (finite verb = :final)						
type	intro	end	type	intro	end	tense
REL-S	der/die/das, welcher/e/es	,	INH-S-ANF	–	,	
KAU-S	da, weil, weshalb, zumal		INH-S-ZU	–		INFINITIV+ZU
KONS-S	dass, so dass, um zu, als daß		INH-S-DASS	daß		
KONZ-S	obgleich, obwohl, wenn auch		INH-S-ALS	als		
TEM-S	sobald, nachdem, wenn, bevor		INH-S-WENN	wenn		
KOND-S	wenn, falls, sofern		INH-S-ALSOB	als ob		
MOD-S	indem, dardurch daß, so daß		INH-S-OB	ob		
FIN-S	damit, daß, auf daß, um zu		INH-S-WIE	wie		
AUS-P	wobei, was, nur daß, insofern		INH-S-WH	WH phrase		

Table 26: Inflection Attributes

category	attribute values	description
person (人称)	1	first person (一人称)
	2	second person (二人称)
	2a	form of address (二人称敬称)
	3	third person (三人称)
case (格)	NOM	nominative case (一格)
	GEN	genitive case (二格)
	DAT	dative case (三格)
	AKK	accusative case (四格)
number (数)	SG	singular (単数)
	PL	plural (複数)
gender (性別)	MAS	masculine (男性)
	FEM	feminine (女性)
	NTR	neuter (中性)
comparison (階級)	POS	positive (原級)
	KOM	comparative (比較級)
	SUP	superlative (最高級)
tense (時相)	PRAESENS	present (現在)
	IMPERFEKT	imperfect (過去)
	PERFEKT	perfect (現在完了)
	PLUSQUAMPERFEKT	past perfect (過去完了)
	FUTUR-1	future 1 (未来)
	FUTUR-2	future 2 (未来完了)
	INFINITIV	infinitive (不定詞)
	INFINITIV+ZU	infinitive (不定詞)+"zu"
	PARTIZIP-PRAESENS	present participle (現在分詞)
	PARTIZIP-PERFEKT	perfect participle (過去分詞)
	KONJUNKTIV-1	subjunctive (接続法-1)
	KONJUNKTIV-2	subjunctive (接続法-2)
mood (ムード)	INDIKATIV	indicative (直説法)
	KONJUNKTIV	subjunctive (接続法)
	IMPERATIV	imperative (命令法)
voice (態)	AKTIV	active (能動態)
	PASSIV	progressive passive (受動態)
	ZS-PASSIV	adjectival passive (状態受動)

Table 27: Part-of-Speech Inflection

part-of-speech	attribute key	attribute values
VERB HILFSVERB MODALVERB	:TENSE	PRAESENS IMPERFEKT PERFEKT PLUSQUAMPERFEKT FUTUR-1 FUTUR-2 KONJUNKTIV-1 KONJUNKTIV-2 INFINITIV INFINITIV+ZU PARTIZIP-PRAESENS PARTIZIP-PERFEKT
	:NUMBER	SG PL
	:PERSON	1 2 2a 3
	:MOOD	INDIKATIV KONJUNKTIV IMPERATIV
	:VOICE	AKTIV PASSIV ZS-PASSIV
NOMEN	:GENDER	MAS FEM NTR
	:NUMBER	SG PL
	:CASE	NOM GEN DAT AKK
ADJEKTIV	:COMPARATION	POS KOM SUP
	:ARTICLE	OHNE BESTIMMT UNBESTIMMT
	:GENDER	MAS FEM NTR
	:NUMBER	SG PL
	:CASE	NOM GEN DAT AKK
	:ATTRIBUTIVE-USED-P	NIL T
POSSESSIV-PRONOMEN	:TYPE	ARTIKELWORT SUBSTANTIVWORT
	:GENDER	MAS FEM NTR
	:NUMBER	SG PL
	:CASE	NOM GEN DAT AKK
PERSONAL-PRONOMEN	:PERSON	1 2 2a 3
	:NUMBER	SG PL
	:CASE	NOM GEN DAT AKK
	:GENDER	MAS FEM NTR
REFLEXIV-PRONOMEN	:PERSON	1 2 2a 3
	:NUMBER	SG PL
	:CASE	DAT AKK
ARTIKEL, RELATIV-PRONOMEN	:GENDER	MAS FEM NTR
	:NUMBER	SG PL
	:CASE	NOM GEN DAT AKK
	:DET	DET T
DETERMINATIV, DETERMINATIV-INDEF	:GENDER	MAS FEM NTR
	:NUMBER	SG PL
	:CASE	NOM GEN DAT AKK
ORDINALZAHL	:ARTICLE	OHNE BESTIMMT UNBESTIMMT
	:GENDER	MAS FEM NTR
	:NUMBER	SG PL
	:CASE	NOM GEN DAT AKK
	:ATTRIBUTIVE-USED-P	NIL T
no inflection	ADVERB, BUCHSTABE, FRAGEADVERB, INTERROGATIV-PRONOMEN, INITIAL, KARDINALZAHL, KOORD-KONJUNKTION, PRAEPOSITION, PARTIKEL, SUBORD-KONJUNKTION, UNIT VERBZUSATZ, STOP-WORD	

Top-level Functions

g–generation *transfer-result* function

Use: This function forms the main interface between the transfer and the German generation module of the TDMT system.

Arguments: *transfer-result* is the result of the transfer analysis of the Japanese input sentence. It consists of an encapsulated list containing part-of-sentence specifications and forms the input of the generation process.

Return: The generated sentence string.

Example: (1) *main clause, with leading subord-clause:*

```
((SATZGEFUEGE KAU-S (INTRO ”da”) (SUB (TIME (ADVERB ”heute”)))
                          (*HILFSVERB ”sein”) (NOMEN ”wochenende”))
 (*HILFSVERB ”sein”) (*SUB ”es”) (SUB (NOMEN ”preis”))
 (*HILFSVERB ”sein”) (*SUB ”es”) (HILFSVERB ”werden”)
 (ADJEKTIV ”teuer” :comparation KOM))
```

⇓

”Da heute Wochenende ist, wird der Preis teurer.”

(2) *question:*

```
((FRAGE (*SUB ”man”) (SUB (NOMEN ”einzelzimmer”)
       (PP ”mit” (NOMEN ”bad”))) (*HILFSVERB ”sein”) (*SUB ”es”)
       (WH ”wie teuer”)))
```

⇓

”Wie teuer ist das Einzelzimmer mit Bad?”

(3) *recursive subord-clauses:*

```
((FIX-INTRO ”bitte?”) (*SUB ”es”) (SUB (NOMEN ”gebühr”)
 (SATZGEFUEGE REL-S
   (INTRO (RELATIVPRONOMEN ”der/die/das”)) (*SUB ”ich”)
   (ATTR (VERB ”sehen”) :TENSE PERFEKT)
   (ATTR (PP ”in” (NOMEN ”reiseführer”)) :CASE DAT)))
 (HILFSVERB ”sein”) (FIX-INTRO ”ich glaube,”) (VP ”anders sein”))
```

⇓

”Bitte? Es scheint, daß der Preis, den ich im Reiseführer gesehen habe, anders ist.”

preprocess *transfer* — function

Use: Determine the type of the input sentence and check syntax of the generation input.

Arguments: *transfer* is the result of the transfer analysis of the Japanese input sentence. It consists of an encapsulated list containing part-of-sentence specifications and forms the input of the generation process.

Return: If syntax is correct, the modified (first symbol=type of sentence) list-structure will be returned.

Remarks: If no sentence type is specified, i.e. the first list-element is not a sentence type, we have to search for a main clause (a list, whose first element contains a main clause type) or single main clause type symbol.

In the case of a question (FRAGE) the type of the question (YN–Q, WH–Q) has to be chosen. WH-Q is assigned, if the sentence contains a WH phrase, else YN–Q is chosen.

However, if the first element of the transfer result represents a subordinated clause, the type SATZ_VERB-FIRST, i.e. a main clause preceded by a subordinated clause will be used.

If no type specification is found, the default-type SATZ will be assigned.

Example: (1) *main clause:*

```
((INTRO "ja,") (*SUB "wir") "können" (*AKK-OBJ "es") (VERB "besorgen"))
                                  ⇓
(SATZ (INTRO "ja,") (*SUB "wir") "können" (*AKK-OBJ "es") (VERB "besorgen"))
```

(2) *question:*

```
(FRAGE (SUB "sie") (HILFSVERB "sind") (WH "wieviele personen"))
                          ⇓
(WH–Q (SUB "sie") (HILFSVERB "sind") (WH "wieviele personen"))
```

(3) *main clause, with leading subord-clause:*

```
((SATZGEFUEGE KAU-S (INTRO "da") (SUB (TIME (ADVERB "heute")))
                          (*HILFSVERB "sein") (NOMEN "wochenende"))
    (SUB (NOMEN "gebühr")) (*HILFSVERB "sein") (*SUB "es")
  (HILFSVERB "werden") (ADJEKTIV "teuer" :comparation KOM))
                                  ⇓
(SATZ_VERB-FIRST
  (SATZGEFUEGE KAU-S (INTRO "da") (SUB (TIME (ADVERB "heute")))
                    (*HILFSVERB "sein") (NOMEN "wochenende"))
  (SUB (NOMEN "gebühr")) (*HILFSVERB "sein") (*SUB "es")
    (HILFSVERB "werden") (ADJEKTIV "teuer" :comparation KOM))
```

generate *transfer* — function

Use: Internal generation of the modified transfer result.

Arguments: *transfer* is the modified result (sentence type adopted) of the transfer analysis. It consist of an encapsulated list containing part-of-sentence specifications and forms the input of the generation process.

Return: A string, which consists of the substrings of the recursive generated subordinated and coordinated clauses, surrounded by the affiliated punctuation characters.

Remarks: Depending on the analyzed sentence type a internal sentence structure will be created and assigned to the global variable ***simple-sentence***. The analysis of the sentence components will be added to this structure. Subordinated and coordinated sentence will be generated recursively, whereby each substructure is updated to the global variable ***complex-sentence***.

Example: (1) *main clause:*

```
((INTRO ”ja,”) (*SUB ”wir”) ”können” (*AKK-OBJ ”es”) (VERB ”besorgen”))
                                   ⇓
                    ”.ja , wir können es besorgen.”
```

(2) *main clause, with leading subord-clause:*

```
(SATZ_VERB-FIRST
 (SATZGEFUEGE KAU-S (INTRO ”da”) (SUB (TIME (ADVERB ”heute”)))
                    (*HILFSVERB ”sein”) (NOMEN ”wochenende”))
 (SUB (NOMEN ”gebühr”)) (*HILFSVERB ”sein”) (*SUB ”es”)
 (HILFSVERB ”werden”) (ADJEKTIV ”teuer” :comparation KOM))
                                   ⇓
              ”.,da heute Wochenende ist, wird der Preis teurer.”
```

(3) *question:*

```
          (WH-Q (*SUB ”sie” :PERSON 2A) (*HILFSVERB ”sind”)
                        (WH ”wieviele personen”))
                                   ⇓
                       ”?wieviele Personen sind Sie?”
```

(4) *co-ordinated questions:*

```
 (YN-Q (*SUB ”es”) (*HILFSVERB ”sein”) (SUB ”sie” :PERSON 2A)
   (VERB ”haben” :TENSE KONJUNKTIV-2)
(AKK-OBJ (NOMEN ”doppelzimmer”)) (VERBZUSATZ ”gerne”)
   (SATZREIHE (INTRO ”oder”) YN-Q
              (NP UNBESTIMMT (NOMEN ”zweibettzimmer”))))
                                   ⇓
    ”?hätten Sie gerne ein Doppelzimmer ? oder ein Zweibettzimmer ? ?”
```

inflect *SenPart-analyzed* function

Use: Generate inflection forms of the analyzed sentence elements.

Arguments: *SenPart-analyzed* is the result of the analysis of a sentence part, i.e. a list of the form: $(((\mathit{PartOfSp}_j\ \mathit{stem}_j\ (:\ \mathit{key}_{i1}\ \mathit{value}_{i1}\ \ldots)_i\ )_j\ )_k\ )$

Return: A string, which represents the inflection of the part-of-sentence given the first specified inflection.

Remarks: These inflection function is applied to all part of the sentence, which are assigned in the current sentence structure. All parts are concatenated (delimiter: " ").

Example:

```
(1)                 (INFLECT (((ADVERB "heute"))))
                                  ⇓
                               "heute"

(2)        (INFLECT (((VERB "reservier" (:TENSE INFINITIV)))))
                                  ⇓
                            "reservieren"

(3) (INFLECT (((PERSONALPRONOMEN "ich" (:CASE NOM :PERSON 1
                                        :NUMBER SG)))))
                                  ⇓
                                "ich"

(4) (INFLECT (((ADJEKTIV "teuer" (:ATTRIBUTIVE-USED-P NIL
                                  :COMPARATION KOM)))))
                                  ⇓
                               "teurer"

(5) (INFLECT (((MODALVERB "mo~g"
                 (:TENSE PRAESENS :NUMBER SG :PERSON 1
                  :MOOD INDIKATIV :VOICE AKTIV)
                 (:TENSE IMPERFEKT :NUMBER SG :PERSON 1
                  :MOOD INDIKATIV :VOICE AKTIV) ...))))
                                  ⇓
                              "möchte"

(6) (INFLECT (((DETERMINATIV-INDEF "ein"
                 (:GENDER NTR :NUMBER PL :CASE AKK)
                 (:GENDER NTR :NUMBER SG :CASE AKK)))
               ((NOMEN "zimmer"
                 (:GENDER NTR :NUMBER PL :CASE AKK)
                 (:GENDER NTR :NUMBER SG :CASE AKK)))
               ((PRAEPOSITION "fu~r")) ((ADVERB "heute"))
               ((NOMEN "nacht"
                 (:GENDER FEM :NUMBER SG :CASE AKK)))))
                                  ⇓
                      "ein Zimmer für heute Nacht"
```

inflect–internal *PartOfSp stem*
&key *tense mood voice gender case person number*
comparation type attributive-used-p article det function

Use: Internal inflection of a single part-of-speech.

Arguments:

PartOfSp	symbol, specifying a part-of-speech.
stem	string, representing a canonical word-stem.
tense	**PRAESENS** IMPERFEKT PERFEKT FUTUR-1 FUTUR-2 PLUSQUAMPERFEKT KONJUNKTIV-1 KONJUNKTIV-2 INFINITIV INFINITIV+ZU PARTIZIP-PERFEKT PARTIZIP-PERFEKT PARTIZIP-PRAESENS PARTIZIP-PRAESENS-MIT-ZU
mood	**INDIKATIV** KONJUNKTIV IMPERATIV
voice	**AKTIV** PASSIV ZS-PASSIV
gender	MAS FEM **NTR**
case	**NOM** GEN DAT AKK
person	1 2 2a **3**
number	**SG** PL
comparation	**POS** KOM SUP
type	**ARTIKELWORT** SUBSTANTIVWORT
attributive–used–p	**NIL** T PRAEDIKATIV-GEBRAUCHT
article	**OHNE** BESTIMMT UNBESTIMMT
det	**DET** T

Return: A single string or a list of strings, e.g. in the case of compound tenses, multiple strings will be generated.

Remarks: This function forms the interface to the inflection functions defined in the MORPHIX submodule (cf. Appendix C).

If one of the key parameters is not specified in the parameter list of the function call, the default value (marked as **boldface**) will be used.

Example:

(INFLECT-INTERNAL MODALVERB "mo˜g" :PERSON 1 :TENSE KONJUNKTIV-2)
⇓
"möchte"

(INFLECT-INTERNAL NOMEN "zimmer" :GENDER NTR :CASE AKK :NUMBER SG)
⇓
"Zimmer"

re–transform–umlaut *word* &optional *PartOfSp* function

Use: Re-transformation of the internal representation of diacritic characters (e.g. the German umlaut) into the written language form.

Arguments: *word* string, whose "Umlaut" has to be re-transformed.

PartOfSp part-of-speech of the specified word.

Return: The re-transformed string.

Remarks: The re-transformation table is given below:

aˆ → â	a` → à	a´ → á	a˜ → ä	Aˆ → Â	A` → À	A´ → Á	A˜ → Ä
eˆ → ê	e` → è	e´ → é	e˜ → ë	Eˆ → Ê	E` → È	E´ → É	E˜ → Ë
iˆ → î	i` → ì	i´ → í	i˜ → ï	Iˆ → Î	I` → Ì	I´ → Í	I˜ → Ï
oˆ → ô	o` → ò	o´ → ó	o˜ → ö	Oˆ → Ô	O` → Ò	O´ → Ó	O˜ → Ö
uˆ → û	u` → ù	u´ → ú	u˜ → ü	Uˆ → Û	U` → Ù	U´ → Ú	U˜ → Ü
cˆ → ç	C` → Ç	s˜ → ß					

Example: (RE-TRANSFORM-UMLAUT "Reisefu˜hrer" NOMEN) ⇒ "Reiseführer"

(RE-TRANSFORM-UMLAUT " Cafe'" NOMEN) ⇒ "Café"

(RE-TRANSFORM-UMLAUT "Fass" NOMEN) ⇒ "Faß"

postprocess *generation* function

Use: Revision of generated surface string for display. Handling of punctuation characters and orthographic modifications)

Arguments: *generation* is the result of the generation process (string).

Return: Revised surface string and the modified part-of-speech list

Remark: The multiple inserted punctuation markers caused by complex sentence structures are adopted (selecting last one and upcase succeeding word).

Example: (POSTPROCESS ".,da heute Wochenende ist, wird der Preis teurer.")

⇓

"Da heute Wochenende ist, wird der Preis teurer."

(POSTPROCESS "?ziehen Sie ein Doppelzimmer vor ?oder ein Zweibettzimmer ??")

⇓

"Ziehen Sie ein Doppelzimmer vor? Oder ein Zweibettzimmer?"

G-FILTER-input

global variable

Use: Interface between TDMT (translation) and CHATR (synthesis): providing complete linguistic knowledge of translation and phonetic word information for use in CHATR's prosody module.

Remarks: For each translation constituents (phrase) the linguistic information is specified in one line consisting of 4 fields separated by "|":

1. surface words (list of words separated by spaces)
2. linguistic information consisting of 3 sub-fields separated by ":" (sentence type, topological field, list of part-of-speeches of field (1))
3. syllabified phonetic information (syllable marker '-') of field (1)
4. stress pattern information (0-no stress, 1-stress) of field (1)

Example:

```
"bitte|SATZ:INTRO:FIX-INTRO|b.I.-t.@.|10
?|SATZ:BOUNDARY-END:INTERPUNKTION|4|0
es scheint|SATZ:INTRO:FIX-INTRO FIX-INTRO|@.s. S.aI.n.t.|0 1
,|SATZ:BOUNDARY:INTERPUNKTION|2|0
die gebühr|SATZ:SUBJECT:DETERMINATIV NOMEN|d.i:. g.@.-b.y:6.|1 01
ist|SATZ:V-FIN:HILFSVERB|I.s.t.|1
anders|SATZ:VP-ADVERB:ADVERB|a.n.-d.@.r.s.|10
.|SATZ:BOUNDARY-END:INTERPUNKTION|4|0
"
```

debug

global variable

Use: Enable (value=T) or disable (value=NIL) the output of debug messages during the generation process.

Remarks: Default value is NIL, in order to avoid warnings during the translation process of the TDMT system.

update

global variable

Use: Enable (value=T) or disable (value=NIL) the update of the results of the analysis process.

Remarks: This global variable is used to temporarily disable the update of the analyzed sub-parts during the recursive analysis of a part-of-sentence. If the update would not be disabled the analysis result will appear twice in the generation result, first because of the single part-of-speech analysis and second within the analyzed part-of-sentence.

sentence structure

Use: The main sentence structure representing the analyzed information.

Slots:

type	sentence type (cf. Table 25)
transfer	transfer result (list of target expressions)
preprocess	preprocess result (list of target expressions)
generation	generation result of current and all subordinate utterances (string)
punctuation	”.” ”!” ”?” ”,”
negation	flag for sentence negation
V-fin-pos	:first :second :final
person	finite verb attribute (cf. Table 26)
number	finite verb attribute (cf. Table 26)
tense	finite verb attribute (cf. Table 26)
mood	finite verb attribute (cf. Table 26)
voice	finite verb attribute (cf. Table 26)
topo	topological field structure of current sentence type
sepa	analyzed sentence-elements
intro	analyzed sentence-elements
F-pre	analyzed sentence-elements
V-fin	analyzed sentence-elements
F-middle	analyzed sentence-elements
V-inf	analyzed sentence-elements
F-post	analyzed sentence-elements
subject	analyzed sentence-elements
nom-object	analyzed sentence-elements
gen-object	analyzed sentence-elements
dat-object	analyzed sentence-elements
akk-object	analyzed sentence-elements
pp	analyzed sentence-elements
time	analyzed sentence-elements
place	analyzed sentence-elements
ref-verb-pron	analyzed reflexive-pronoun
VP-non-verb	analyzed sentence-elements
VP-adverb	analyzed sentence-elements
flag-sub	NIL SUB *SUB
flag-nom-obj	NIL NOM-OBJ *NOM-OBJ
flag-gen-obj	NIL GEN-OBJ *GEN-OBJ
flag-dat-obj	NIL DAT-OBJ *DAT-OBJ
flag-akk-obj	NIL AKK-OBJ *AKK-OBJ
flag-verb	NIL VERB MODALVERB HILFSVERB *VERB *MODALVERB *HILFSVERB
hokan	list of HokanMarker and assigned verb part-of-speech