Institut für deutsche Sprache, Mannheim, Germany

MLAP93-21 MECOLB

Deliverable D5

WP2 - Lemmatizer

Final Report, July 1994

Task 2.1: Lemmatizer

by Cyril Belica

Index

Introduction

This report describes the results achieved in the Work Package 2 in the development of a German lemmatizer. The work has been done in three phases. In the early stage of work we have focused on a detailed specification of the goals, on the overall design of the system, and especially on the algorithmic aspects of the procedure.

In the second phase a pilot implementation has been coded to enable quality assessment feed back as soon as in the middle of the work package term. Also, further analysis of relevant linguistic phenomena and an estimate of the computational complexity have been carried out.

In the final period of time the release version of the lemmatizer has been implemented and tested by lemmatizing a 35 million German corpus in Mannheim.

User Needs

One of the basic problems with accessing text corpora of languages with complex inflection, derivation and word composition morphology is the number of different word forms related to a single lemma. To retrieve all relevant contexts for a word, several methods are currently used.

For instance, the user can keyboard and search all possible inflected forms of the word and merge the results of the partial searches. This approach is a real nuisance if there are tens or hundreds of inflected forms, and becomes unfeasible as soon as e.g. German word composition (*Wortbildung*) in general is concerned, since the set of **possible** word forms related to a single lemma is open-ended due to the numerous productive word composition patterns.

Another frequent method to cope with the morphological transparency is to use wild card searching. This approach works quite well if there are enoung distinctive characters in the search pattern preventing it from matching random substrings in longer words. With shorter or extremly frequent patterns, and especially in languages with a high average word length (e.g. German), the percentage of irrelevant random hits rises exponentially.

The best results accessing raw German texts in a morphologically transparent way can be achieved by a combination of the previous methods. But even these results are rather disappointing.

The only approach that seems to work for German is to make use of the linguistic annotation of the corpus, if provided. Unfortunatelly, only very few of the currently available German corpora include annotations necessary to help resolve the queries mentioned above.

Design Principles

To cope with this problem a German lemmatizer has been included in MECOLB. In the COSMAS-II framework the lemmatizer is not intended as a regular annotation generator, but rather as a low-cost German instance of a general slot in the **basic access method** providing a robust morphologically transparent word form access to corpora in which no satisfactory lemma and/or word class annotation is available.

The following figure gives an overview of how the lemmatizer will be slotted into the COSMAS-II corpus maintenance system. In general, one or more lemmas are generated for each word form during input text processing and passed to the indexing engine. Along with the text words, the lemmas are included in the internal structures of the the low-level word access mechanism.

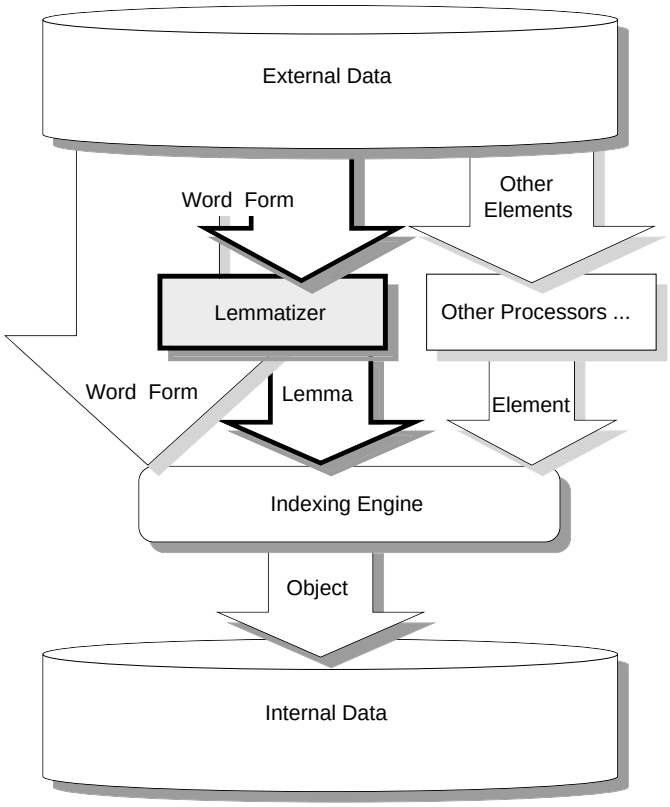

Fig.1 The Lemmatizer Slot in MECOLB

Bearing in mind the limited man-power for the development, the following design principles have been adopted:

- The lexical and morpho-syntactic knowledge needed in the analysis will be extracted from the SDW-Lexicon **[SON1980]**.
- The lemmatizer will operate on the single word level. No word context will be investigated.
- To minimize the negative consequences of the previous decision (the lack of context-based information for the disambiguation), probabilistic and heuristic features will be included in the lemmatizer.
- The lemmatizer will perform word form analysis rather than word form generation for a given lemma.
- The lemmatizer will be modular and can also be used in other operating environments, e.g. as a simple *morpho-server*.
- The procedure will be robust enough to lemmatize real-life German text corpora.

Implementation

In this chapter the major modules and the algorithmic aspects of the procedure are briefly described. The following figure outlines the general structure of the lemmatizer.

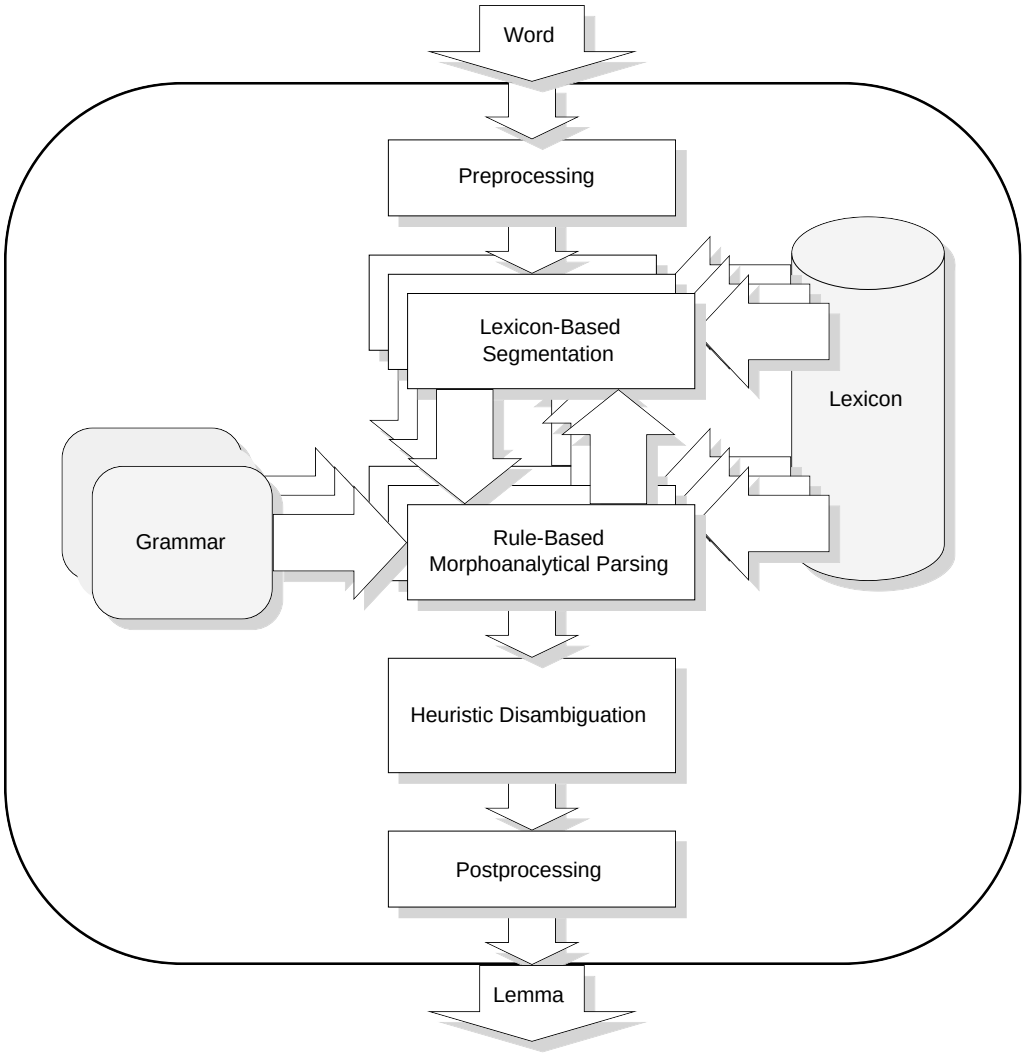

Fig.2 The Structure of the Lemmatizer

Preprocessing

The goal of preprocessing is to perform tasks not directly related to the lemmatization problem, e.g. to eliminate word forms from the running text, to cope with hyphenation, to normalize the *Umlaut*-character coding, etc. The

preprocessor is implemented as a filter in order to allow for different input text formats.

Lexicon

The lexicon used in the lemmatizer has been constructed by extracting information from the *Saarbrücker Deutsches Wörterbuch* [SON1980] and by inserting entries for new lexical items. The lexicon contains at present 161.411 entries giving the morphological and morpho-syntactic description of the lexemes.

During the project term the lexicon has been augmented with simple probabilistic information which is evaluated in the process of disambiguation.

For each available lexeme the lexicon contains one or more entries with alternative morpho-syntactic paradigms (e.g. *er* as a pronoun, as a verb prefix, as an inflection suffix of nouns, as an inflection suffix of adjectives, as an adjective comparative morpheme and as a linking morpheme in word composition).

For historical reasons all entries are given in lowercase and all *Umlaut* characters are substituted by their international transcription (i.e. *ae, ue, ss,* etc.).

Segmentation

The segmentation of the input word forms is performed by a lexicon-driven bottom-up algorithm for morpheme recognition according to the longest-match-first strategy. The algorithm uses a complete backtracking to generate an exhaustive segmentation tree for each input word form. The nodes of the segmentation tree are linked to the morpho-syntactic paradigms of the isolated word constituents in the lexicon.

The variant segmentations within the tree reflect the different breakdown possibilities of the input string as well as the competitive linguistic interpretations of a particular breakdown in terms of the paradigms of its constituents.

In a refinement work step, the segmentation algorithm has been augmented in order to handle consonant clustering at morpheme boundaries (e.g. *Schi**ff**ahrt*) and grapheme mutation in hyphenated words (e.g. *schlu**k-k**en*).

We have also included in the process of segmentation probabilistic information giving the degree of affinity of particular morphemes with their neighbours in respect to word composition.

Grammar and Parser

The bottom-up constructed segmentation trees are parsed by a morpho-analytical parser in order to discard all ill-formed segmentation readings. The parser (2500 entries, 250 states) generated by a standard UNIX *yacc* utility operates on a set of

about 180 rules for inflection, derivation and word composition morphology of German (see chapter Linguistic Background).

Over 60 additional constraints are used to validate the gender, number and case dependencies, the consistency of affixes and linking morphemes, etc. These constraints are implemented in form od C-functions returning status to the parser.

The parser also assigns weights to every accepted segmentation reading based on the parse tree used and on the affinity information provided by the segmentation algorithm.

Context Switching

To control the process of segmentation and parsing a context switching driver has been implemented. The driver interrupts the segmentation algorithm after a new branch has been added to the segmentation tree and then invokes the morpho-analytical parser in an interject mode. The parser checks the interim conformancy of the generated partial reading with the grammar and returns to the driver.

The driver then either discards an ill-formed segmentation subtree and re-enters the segmentation algorithm with a *backtrack condition*, or signals a *proceed condition* and restores the held state of the segmentation process.

Finaly, for every completed segmentation of a word form the driver invokes the parser in a pedantic mode in order to validate the current reading including all constraint rules. Successfuly parsed segmentation trees are assigned weights and pushed on a stack for subsequent disambiguation. Ill-formed readings are discarded. In either case a *backtrack condition* is signaled and the context is switched back to the segmentation process. The segmentation will be continued to search out the remaining readings of the word form.

Disambiguation

The most challenging aspect of designing a German lemmatizer seems to be the tuning of the way how the "correct" reading(s) are determined among those that have passed the morpho-analytical parsing. An exhaustive segmentation breakdown does not, of course, yield directly usable results.

> EXAMPLE 1. Given the word form *Gehalt*, the analysis *geh-(gehen)* and *Alt(singing voice)* would be a mistake, although it follows one of the most productive word composition patterns in German, namely **<verb-present-tense-stem>+<noun>** (cf. *Gehweg*, etc.).

The process of making decisions when looking for the **ultimate** breakdown **constituents** of a compound German word is called disambiguation in this report. This usage covers neither a word class nor a part of speech disambiguation.

What is actually done here is an attempt to rank the accepted readings on the tree stack in the order of decreasing probabilities, to pop the less-preferred ones and,

finaly, to select the lemma names. This is done by heuristically quantifying the "morphological distance" between the input word form and the breakdown constituents for a particular segmentation tree. The key role here is played by the degree of (semantically loaded) lexicalization of composite constituents of that particular breakdown.

Various heuristics have been studied to optimize the disambiguation results up to this point. Among others, the following precedence of linguistic phenomena are taken into account: base form, inflection, copulative word composition, affixing, word derivation, elliptical word composition. Within a single precedence level the following word class priorities are used: noun, adjective, verb, adverb, others.

> EXAMPLE 2. Let us assume that there are two different segmentation trees for a given word form. In the first reading, the parse tree includes some word class derivation rule. The second segmentation tree, however, parsed 'much more easily', e.g. only inflection rules have been applied to prove its consistency. Thus the second reading will be assumed to be more probable than the first one.

> EXAMPLE 3: The word form *Allergiekranke* parsed as *aller-Giek-Ranke* (cf. *Allerweltsanklage*) vs. *Allergie-krank-e*. The first reading ranks worse due to the statistically rare constituent *Giek*.

Given the ranked stack of possible segmentation trees the disambiguator discards all readings that ranked significantly worse than the best one. The significance level used in this step is a tunable parameter.

In the next step, the disambiguator extracts the lemmas from the purged segmentation tree. The extraction is lexicon-driven, i.e. the disambiguator links the input word form to the lexemes corresponding with the breakdown constituents.

Postprocessing

After the disambiguation is completed, the internal morpho-analytical information collected by the parser about the input word form is reduced to the following surface structure:

<word_form>

<composition_tag><derivation_tag><lemma_name>

The output file (see option **-b** in User's Reference) contains two types of lines. If the first character in the line is not the tabulator character, the line contains a copy of the input <word_form>. Lines of the second type begin with a single tabulator character and list tagged lemmas of the preceeding <word_form>. The <composition_tag> and <derivation_tag> are one-byte tags indicating whether or nor the word composition and/or word derivation rules have been used during the analysis to reduce the <word_form> to the <lemma_name>. Both tags use the

plus sign to indicate the presence and the underscore character to indicate the absence of the property tagged. All combinations of tag values are possible.

EXAMPLE 1: Let us assume that the output of the lemmatizer contained the following lines:

höherem

_ _ *hoch*

Spielplätze

_ _ *Spielplatz*

\+ _ *Spiel*

\+ _ *Platz*

Beeinträchtigung

_ _ *Beeinträchtigung*

_ + *beeinträchtigen*

_ + *-ung*

To reduce the word form *höherem* to the lemma name *hoch*, only comparative and inflection rules have been used. Consequently, both tags of the lemma name *hoch* are negative. The lemma name *Platz* has a positive value of the <composition_tag> in respect to the input word form *Spielplätze*. The lemma name *beeinträchtigen* has a positive value of the <derivation_tag> due to the mismatch in word class with the input word form *Beeinträchtigung*. Note the special lemma name *„-ung“* standing for the suffix *„ung“*. Similarily, prefix lemma names have the form „prefix-“, e.g. *„zer-“*.

The rest of the internal morpho-analytical information collected by the parser about the input word form during the analysis is discarded, though it could be used by other applications in subsequent analytical steps. After discarding the paradigms of the breakdown constituents the readings with detected homography or polysemy are assigned the same lemma name.

EXAMPLE 2: During the analysis, the word form *seiner* is assigned the following lemmas:

Reading	**Lemma Name**	**Paradigm**
1.	*er/es*	*personal pronoun, etc.*
2.	*sein*	*possessive, etc.*

etc.

The word *wäre* is assigned a single lemma:

Reading	**Lemma Name**	**Paradigm**
1.	*sein*	*verb, etc.*

After discarding the morpho-syntactic paradigms linked to the above readings, the both analyzed word forms are assigned **the same** lemma name, namely *sein.*

Software notes

All program modules of the lemmatizer are written in ANSI C. The parser is generated by a standard UNIX *yacc* utility. On a 25MHz Motorola-68100 processor running DG/UX 5.4, the program requires about 5MB of main memory and lemmatizes a 915.841 items long word list derived from a 35 million real-life German corpus at the speed of about 150 entries per second.

Linguistic Background

Word Constituent Classes

The lemmatizer makes use of the following word constituent classes to analyze the morphological structure of the input word forms:

1.	VRB	verb
2.	VRB_STARR	the verb *sein*
3.	ADJ	adjective
4.	ADJNEG	negation prefix of adjectives
5.	ADJSTEIG	comparative suffix
6.	ADJFLEX	adjective inflection suffix
7.	VRB_FLEX	verb inflection suffix
8.	VRB_INFLEX	verb infinitive suffix
9.	VRB_PIFLEX	verb present participle suffix
10.	SUB_FLEX	general noun inflection suffix
11.	SUB_END_EN	noun inflection suffix *en*
12.	SUB_END_E	noun inflection suffix *e*
13.	SUB_END_N	noun inflection suffix *n*
14.	SUB_END_S	noun inflection suffix *s*
15.	SUB_END_ER	noun inflection suffix *er*
16.	SUB_END_SE	noun inflection suffix *se*
17.	SUB_END_TA	noun inflection suffix *ta*
18.	SUB_END_IEN	noun inflection suffix *ien*
19.	SUB_END_NEN	noun inflection suffix *nen*
20.	SUB_END_TE	noun inflection suffix *te*
21.	SUB_END_TEN	noun inflection suffix *ten*
22.	SUB_END_ES	noun inflection suffix *es*
23.	SUB_END_SES	noun inflection suffix *ses*
24.	SUB_END_NS	noun inflection suffix *ns*
25.	SUB_END_ENS	noun inflection suffix *ens*
26.	SUB_AMBI	noun stem

27.	SUB_SING	singular noun stem
28.	SUB_PLUR	plural noun stem
29.	FUGE_E	linking morpheme *e*
30.	FUGE_EN	linking morpheme *en*
31.	FUGE_N	linking morpheme *n*
32.	FUGE_ER	linking morpheme *er*
33.	FUGE_S	linking morpheme *s*
34.	FUGE_ES	linking morpheme *es*
35.	LEERE_FUGE	empty linking morpheme
36.	ADJ_POSIT	positive adjective stem
37.	ADJ_ALLE	unspecific adjective stem
38.	ADJ_KOMP	comparative adjective stem
39.	ADJ_SUPER	superlative adjective stem
40.	ADJ_STEIG	comparative/superlative adjective stem
41.	ADJ_STEIG_ER	comparative morpheme *er*
42.	ADJ_STEIG_ST	superlative morpheme *st*
43.	ADJ_STEIG_EST	superlative morpheme *est*
44.	PREFIX_VSA	noun, verb or adjective prefix
45.	SUFFIX_VRB	verb derivation suffix
46.	SUFFIX_SUB	noun derivation suffix
47.	SUFFIX_ADJ	adjective derivation suffix
48.	FUNKW	function word
49.	ADVERB	adverb
50.	KONJUNKT	conjunction
51.	PRAEPOSIT	preposition
52.	ZAHLWORT	numeral primitive
53.	ZAHLKLEBER	numeral composition morpheme
54.	FREMDWORT	foreign
55.	VRBZUSATZ	verb specific prefix
56.	ZU	morpheme *zu*
57.	EIGENNAME	proper name
58.	HYPHENATED	eliptical hyphen
59.	AUX	unknown

Subclassification of verbs

1. infinitive stem
2. special present tense stem
3. past participle stem
4. infinitive stem containing *zu*
5. *Präteritum*-stem
6. subjunctive stem

Morphological Patterns

This section gives a simplified informal overview of the inflection, derivation and word composition patterns handled by the lemmatizer. Each paragraph in this section defines one morphological **token** (printed in underscored boldface). The token defined is followed by a line-by-line list of definitions (a rule set) consisting of token sequences. Each definition line (a rule) is a single alternative of how the token can be constructed. The token **word_form** is the start symbol of the grammar. Tokens in uppercase are terminal symbols retrieved from the lexicon. Lowercase tokens are non-terminals indicating the analysis path of the parser.

The most of the congruency checking for number, gender, case, mode, tense, afixing etc. is performed by procedures linked to each rule set and, consequently, does not appear in the declarative part of the grammar description.

word_form

wf_unpref
or PREFIX_VSA wf_unpref
or attr
or PRONOMIN
or PRAEPOSIT
or KONJUNKT
or FREMDWORT
or FUNKW
or EIGENNAME
or EIGENNAME EIG_END_S

NOTE: Prefix compatibility is also checked in this set of rules.

wf_noattr

adj

or vrb

or attr vrb

COVERAGE EXAMPLE: The third rule covers e.g. *entgegenkommen.*

wf_attr

sub

or sub_1 vrb_1 sub

or sub_1 sub_2 vrb_1 sub

or sub_1 adj

or sub_1 sub_2 adj

or sub_1 sub_2 sub_2 adj

or vrb_1 adj

or sub_1 vrb

or sub_1 sub_2 vrb

or sub_1 sub_2 sub_2 vrb

or adjform vrb

or adjform vrb_1 sub

or vrb_1 sub

or vrb_2 sub

NOTE: This set of rules handles the recursive surface word composition, checks its consistency and assigns weights according to the components found.
COVERAGE EXAMPLES:. *Radrennfahrer, Stabhochsprung, Gemischtwaren, Gemischtwaren-laden, Betonmischmaschine, Schnelllaufschalter, freilaufen, arbeitsgerichtlich, einkommensteuerpflichtig, mengenmäßig.*

wf_unpref

wf_unpref_

or wf_unpref_ HYPHENATED

NOTE: This rules handle eliptical hyphenation.
COVERAGE EXAMPLE: *Zentral- und Südafrika*.

wf_unpref_

wf_noattr

or wf_attr

or attr wf_attr

NOTE: Recursive attribute prefixing.
COVERAGE EXAMPLE: *Nimmersatt, Rundumschlag, Rückfahrkarte, Immergrün.*

sub

sub_1 sub_rest

or sub_4

NOTE: The following rule sets are responsible for parsing noun primitives. Consistency of the linking morphemes is also checked here. Compound elements are assigned new weights.

sub_rest

sub_2 sub_2 sub_2 sub_3

or sub_2 sub_2 sub_3

or sub_2 sub_3

or sub_3

sub_1

sub_1_teil

NOTE: An empty linking morpheme is checked in this rule.

sub_1_teil

sub_4

or sub_4 fuge

sub_2

sub_3

or sub_3 fuge

COVERAGE EXAMPLE: *Holzkohle-anzünder* vs. *Holzkohle-n-anzünder*

sub_3

sub_3_teil

NOTE: An empty linking morpheme is checked in this rule.
COVERAGE EXAMPLE: *Holzschuppen*

sub_3_teil

sub_4

or sub_prefix sub_4

COVERAGE EXAMPLES: *Urwald, Mißachtung, Neopaleonthologie*

sub_4

sub_4_teil

or adjform sub_4_teil

or adjform sub_prefix sub_4_teil

COVERAGE EXAMPLES: E.g. *Braunkohle, Buntfarbstift, Kleinstwagen, Höhereinstufung*

sub_4_teil

sub_sing

or sub_flekt

or sub_adjneg

sub_prefix

PREFIX_VSA

COVERAGE EXAMPLE: *multi-, auto-, pseudo-, rück-, über-,* etc.

sub_flekt

sub_p_stamm

or sub_s_stamm sub_flex_bas

or sub_p_stamm sub_flex_ext

or sub_ambi sub_flex_ext

or vrb_1 SUB_END_S

NOTE: Basic noun inflection.
COVERAGE EXAMPLE: The fourth rule handles e.g. *des Schwimmens*.

sub_sing

sub_s_stamm

or sub_ambi

NOTE: Candidates for a singular noun.
COVERAGE EXAMPLE: *Buch* and *Tisch*, respectively.

sub_ambi

sub_a_stamm

or sub_trans

or sub_s_stamm SUFFIX_SUB

or sub_a_stamm SUFFIX_SUB

NOTE: Nouns with unknown number attribute. Also word derivation without word class transition (noun→noun).
COVERAGE EXAMPLE: *Lehrerin, Metzgerei*.

sub_trans

vrb_1 SUFFIX_SUB

or adjblk SUFFIX_SUB

NOTE: Derivation morphology with word class transition to **nouns**.
COVERAGE EXAMPLE: *Führung, Schönheit*.

sub_adjneg

ADJNEG adjblk SUFFIX_SUB

NOTE: Derivation morphology with word class transition adjective➔noun.
COVERAGE EXAMPLE: This rule covers e.g. *Unsterblichkeit*.

sub_s_stamm

SUB_SING_STAMM

COVERAGE EXAMPLE: *Hand*.

sub_p_stamm

SUB_PLUR_STAMM

COVERAGE EXAMPLE: *Händ-e.*

sub_a_stamm

SUB_AMBI_STAMM

COVERAGE EXAMPLE: *Tisch*.

sub_flex_bas

sub_flex_a

or sub_flex_n

or sub_flex_p

NOTE: Inflexion suffixes are defined here.

sub_flex_ext

sub_flex_bas

or sub_flex_p sub_flex_n

NOTE: Handles the plural dative *-n*.
COVERAGE EXAMPLE: *Übersiedlern*.

sub_flex_a

SUB_END_EN

or SUB_END_TA

or SUB_END_IEN

or SUB_END_NEN

or SUB_END_TEN

or SUB_END_ES

or SUB_END_SES

or SUB_END_NS

or SUB_END_ENS

NOTE: Suffixes with ambiguous number are defined and matched with the stem class.

sub_flex_n

SUB_END_N

sub_flex_p

SUB_END_E

or SUB_END_S

or SUB_END_ER

or SUB_END_SE

or SUB_END_TE

fuge

FUGE_E

or FUGE_EN

or FUGE_N

or FUGE_ER

or FUGE_S

or FUGE_ES

NOTE: Linking morphemes.

adj

adjform

or adjform adj

or attr adj

NOTE: Top rule set for adjectives.
COVERAGE EXAMPLE: The three rules handle e.g. *schön, kleinstmöglich* and *vierfach*, respectively.

adjform

adjblk

or ADJNEG adjblk

NOTE: Basic negation prefixing of adjectives.
COVERAGE EXAMPLE: *unwahrscheinlich*.

adjblk

adjsteig

or adjflekt

or adjsimpl

NOTE: Common node for basic, comparative and inflected adjectives.

adjsteig

adj_tkomp ADJ_STEIG_ER

or adj_tkomp

or adj_alle ADJ_STEIG_ER

or adj_alle ADJ_STEIG_ST

or adj_alle ADJ_STEIG_EST

or adj_tsteig ADJ_STEIG_ER

or adj_tsteig ADJ_STEIG_ST

or adj_tsteig ADJ_STEIG_EST

or adj_tsteig

or adj_tsuper ADJ_STEIG_ST

or adj_tsuper ADJ_STEIG_EST

or adj_tsuper

or vrb_p_l ADJ_STEIG_ER

or vrb_p_l ADJ_STEIG_EST

or vrb_p_l ADJ_STEIG_ST

NOTE: The last three rules handle word class transition from verb present participle.
COVERAGE EXAMPLE: The last rule hadles e.g. *weitgehendst*.

adjflekt

adj_tposit ADJFLEX

or adj_alle ADJFLEX

or adjsteig ADJFLEX

or vrb_p_l ADJFLEX

NOTE: Adjective inflection.
COVERAGE EXAMPLE: *weißem, schmaleres, laufender*.

adjsimpl

adj_tposit

or adj_alle

adj_alle

adj_talle

or adj_trans

adj_trans

sub_sing SUFFIX_ADJ

or sub_p_stamm SUFFIX_ADJ

or vrb_1 SUFFIX_ADJ

NOTE: Defines derivation morphology with word class transitions for adjectives. Consistency of the composition and compound weights are also calculated in this rule set.
COVERAGE EXAMPLE: *lebensfremd, lauffreudig, ereignislos, unaufhaltsam*.

adj_tposit

ADJ_POSIT

NOTE: Positive adjective stems.

adj_talle

ADJ_ALLE

NOTE: Unspecific adjective stems. Weights for word composition are retrieved.

adj_tkomp

ADJ_KOMP

NOTE: Comparative adjective stems.
COVERAGE EXAMPLE: *besser*.

adj_tsuper

ADJ_SUPER

NOTE: Superlative adjective stems.
COVERAGE EXAMPLE: *best*.

adj_tsteig

ADJ_STEIG

NOTE: Comparative or superlative adjective stems. Weights for word composition are retrieved.
COVERAGE EXAMPLE: *läng-er*.

vrb

vrb_1_2

or vrb_1 vrb

NOTE: Recursive top rule set for verbs.

vrb_1_2

vrb_1

or vrb_2

vrb_1

vrb_flekt

or vrb_starr

or vrb_token

COVERAGE EXAMPLE: *gehst, wärst, gehen*.

vrb_2

vrb_p_I

or vrb_p_II

NOTE: Common node for present and past participle.
COVERAGE EXAMPLE: *gehend, gegangen*.

vrb_flekt

vrb_token VRB_FLEX

or vrb_token VRB_INFLEX

NOTE: Basic verb inflection (*Konjugation*). Congruency of person, number and tense is checked here.

vrb_p_I

vrb_token VRB_PIFLEX

NOTE: Present participle.
COVERAGE EXAMPLE: *gehend, aufzuerlegend*.

vrb_p_II

vrb_token ADJFLEX

NOTE: Past participle.
COVERAGE EXAMPLE: abgelegenen.

vrb_token

VRB

or vrbzusatz VRB

or vrbzusatz ZU VRB

or ZU VRB

vrb_starr

VRB_STARR

NOTE: Special word class for *sein*.

vrbzusatz

VRBZUSATZ

COVERAGE EXAMPLE: *miß-, ent-, dar-, zer-, nach-, los-,* etc.

attr

ADVERB

or zahl

COVERAGE EXAMPLE: *draußen, fünfundzwanzig*.

zahl

zahlsimplx

or zahl zahlsimplx

zahlsimplx

ZAHLWORT

or ZAHLWORT ZAHLKLEBER

Conclusion

In the Work Package 2 - Lemmatizer - a portable, efficient and robust program for lemmatizing German words has been developed.

In the preliminary evaluation the lemmatizer was run against a 915.841 items long word list extracted from a 35 million German corpus of written language. We have received a list of 154.536 unparsed entries. Based on a sample of 2.000 entries randomly chosen from this list, the following classification of not lemmatized words is estimated:

	relative frequency within not-lemmatized word list	**relative frequency within total word list**	**relative frequency in 35 million corpus**
non-lexical	2.01%	0.34%	0.01%
proper names	64.18%	10.83%	0.65%
abbreviations	2.61%	0.44%	0.02%
foreign words	16.23%	2.74%	0.01%
spelling errors	12.67%	2.14%	0.01%
German words	2.30%	0.39%	<0.01%
TOTAL NOT-LEMMATIZED	100.00%	16.88%	<0.71%
TOTAL LEMMATIZED		33.76%	>99.29%

Within the successfully parsed entries, some 4.5% are estimated to be assigned in addition to the expected lemma names also some incorrect lemma. An entirely erroneous lemmatization has been observed only in singular cases, predominantly when German morphological analysis is applied to proper names or foreign words.

In the remaining three weeks until the end of the Work Package term further evaluation of the system will be performed aiming at subsequent improvements of the underlying grammar and/or tuning of the disambiguator.

References

SON1980 Sonderforschungsbereich 100 'Elektronische Sprachforschung', Projektbereich A: SALEM. Ein Verfahren zur automatischen Lemmatisierung deutscher Texte. Max Niemeyer Verlag, Tübingen 1980.