GERTWOL
LINGSOFT Oy
QUESTIONNAIRE FOR MORPHOLYMPICS 1994

Name and origin of the participating system

The system is called GERTWOL (German Two-level system) by Lingsoft, Inc. The linguistic description is done in the two-level framework, and is based on the Collins German Dictionary (CGD) regarding the initial contents of the vocabulary. The two-level description can be run on two distinct finite-state implementations, the Lingsoft TWOL program which has evolved from the original first implementation. A second, more advanced implementation in terms of speed and space, is made using the Xerox Lexical tools. Both the framework for describing German morphology and its implementations are independent of the German language which can be seen from the existence of systems made in the same framework using the same tools (Finnish, Swedish, English, Russian, Swahili, Danish, Estonian, French, all with full scale dictionaries).

The GERTWOL is a result of team work: Mariikka Haapalainen as the linguist, Mikko Silvonen as the computer scientist, Krister Lindin directing the work, and Kimmo Koskenniemi and Fred Karlsson as advisors.

1 Conceptual Criteria

1.1 Declarative specification of the lexical entries and the rules

The two-level description consists of a lexicon and rules. The rule component defines morphophonological (or morphographemic) alternations of the language, and the lexicon identifies the morphemes (or other entries used as units) in the description, and their possible sequences in word forms.

The two-level formalism does not dictate in detail which alternations should be treated with the rule component and which by the lexicon. The designer may choose among different styles of description. GERTWOL is intended to be a full scale system with a wide vocabulary from the beginning. It is not an experimental system for exploring various topics of morphology (such as derivation, historical development of the language, etc.).

It would have been fairly easy to implement e.g., many more of the alternations in irregular verbs using rules and morphophonemes. After careful consideration, we decided not to follow that path. Instead, most of the irregular or strictly unproductive alternations have been described without resorting to morphophonemes simply in order to keep the description as transparent and verifiable as possible.
The lexical material behind GERTWOL consists of distinct parts:

1. The bulk of word entries converted from the CGD in an intermediary format (called GLEX format) from which the two-level entries are created by a set of Perl conversion programs.

2. A table for irregular verbs which lists all their stems and possible prefixes.

3. A set of entries for closed class words with idiosyncratic properties (such as pronouns) written as two-level entries (and thus not converted from CGD).

4. Proper nouns and other words not in the CGD which have been added to enhance the coverage of GERTWOL.

The full GERTWOL dictionary is generated from this database automatically using a Unix make program so that only those components will be recomputed where the source has actually changed.

Because of the comprehensive and carefully composed selection of word entries in the Collins German Dictionary, most of the correct derivations of common words are mechanically included in our dictionary. Presence of correct derivations and the absence of incorrect (or overgenerated) derivations is as much true as it is in the CGD (but some corrections have, of course, been made).

In the GLEX file containing cleaned and converted entries from the CGD there is an entry for “maus” (a mouse)

```
f m > .au < s S7 +
```

This is still technically very much like the entry in the file for typesetting. It marks the gender “f”, length “-”, and inflectional class (which has been coded as S7), and a mark for umlaut “+” affecting this word. Only the nouns already in the CGD were ever turned to this format. New nouns are added in a format which is used in the TWOL dictionary.

The above GLEX entry is converted into a TWOL lexicon entry:

```
maus S7+/f;
```

This consists of two fields: “ma_us” is the so-called lexical representation of the stem, and “S7+/f” is the continuation class, which determines the name of the lexicon where the next morpheme (or entry) is to be found. The first field may be omitted if the entry is a null entry with no morphophonological material in it. The lexical representation is related to the surface form using two-level rules. All rules operate in parallel and synchronized.

In most entries the first field is like in the above example. The first field might, however, be divided into two parts by a colon (:) then, the former part is what we like to see in the base form or as features following it (e.g. “+S+FEM+SG+NOM”, “+S+FEM+PL+NOM”). The latter part is, then, the morphophonological representation of a stem or an ending (e.g. “en”, “e@U”). The standard case corresponds to the situation when the parts before and after the colon are identical, i.e. the lexical representation to be found as a part of the analysis is the same as is underlying the base form.

There are minilexicons for inflection etc. which were fixed once the inflectional types have been established. In the following only the entry “maus S7+/f” is specific to the word “maus”. The entries start at the very first lexicon which is called “Root”. Each lexicon starts with a keyword and the name of the lexicon, and this is followed by the entries in the lexicon.

```
LEXICON Root
* SUBST;

LEXICON SUBST
maus S7+/f;

LEXICON S7+/f
S7+/f/end;

SCONT;
s1 SCONT;
CHEN;
LEXICON S7+/f/end
Sg3/f/end;
P11+/f/end;
LEXICON Sg3/f/end
+S+FEM+SG+NOM;
##;
+S+FEM+SG+AKK;
##;
+S+FEM+SG+DAT;
##;
+S+FEM+SG+GEN;
##;
```

"##" marks the next letter as capital, “*” the end of the entry from lexicon SUBST.

"#" marks the long vowel.

! An entry with only one item is a continuation.

! Compound formation.

! Diminutive.

! To singular endings.

! To plural endings.

! Here are (null) endings for singular.

! "##" as a next lexicon name marks the end.

LDV-Forum Bd.11, Nr.1, Jg.1994
The principle in designing this morphological analyzer has been that the CGD has all (or most) necessary distinctions marked, and the distinction is captured from there. The information in the conventional dictionary is related with decisions of the following types:

1. In which lexicon the resulting entry will be placed? This depends e.g., on the part of speech. In some cases, this is based on extensive global computation, e.g., in order to determine compounding characteristics which are expressed partly by placing the entries, and partly by continuation classes. Some entries are put to more than one lexicon.

2. What continuation class is assigned to the entry? This depends on the part of speech, and the set of endings listed in CGD.

3. How should the lexical representation in GLEX be constructed? Some extra markings of the CGD are deleted, some characters are decoded from their multi-character representations, some markings are repositioned, etc.

Rules and associated morphophonemes or diacritic markers are used only where they can be used without exceptions, and where they simplify the combinatorics. The combinatoric properties of various endings in German are more complex than generally expected, and this seems to result in fairly complicated descriptions, no matter how they are designed (if we wish to design an accurate system).

1.2 Relation between lexical entries and word forms

The GLEX entry for "lächeln" (‘smile’) is:
```
el/V la”>.<ch i
```

This has the classification “el/V” which will serve as the continuation class, a multi-character representation “a” for “a”, “>.<” as stress position indicative, and a marking “i” for inseparable (also used for verbs without any prefixes). Again, this technical format is not really meant for users to enter or study. It is used for generating all necessary entries in the TWOL
lexicon. In that respect, the GLEX entry is the non-redundant expression describing the entry of the lexeme, and the TWOL lexicon is something which has been mechanically compiled (and the runtime binary lexicons results of even more comprehensive compilation and preprocessing).

The following are a few analyses of forms of “anlächeln”:

\[
\begin{align*}
\text{anlächeln} & : \text{anlächeln}+V+\text{KONJ}+\text{PRÄS}+\text{SG}3 \\
\text{anlächeln} & : \text{anlächeln}+V+\text{KONJ}+\text{PRÄS}+\text{SG1} \\
\text{anlächeln} & : \text{anlächeln}+V+\text{IND}+\text{PRÄS}+\text{SG1} \\
\text{anlächeln} & : \text{anlächeln}+V+\text{KONJ}+\text{PRÄS}+\text{PL3} \\
\text{anlächeln} & : \text{anlächeln}+V+\text{KONJ}+\text{PRÄS}+\text{PL1} \\
\text{anlächeln} & : \text{anlächeln}+V+\text{IND}+\text{PRÄS}+\text{PL3} \\
\text{anlächeln} & : \text{anlächeln}+V+\text{IND}+\text{PRÄS}+\text{PL1} \\
\text{anlächeln} & : \text{anlächeln}+V+\text{INF} \\
\end{align*}
\]

The analyses are based on the following parts of the lexicon. Again, the entry “lächeln $el/V(sep)$” is the only one pertaining to this particular lexeme. The lexicon combines “an” which is given at the position where ordinary verbs could start, and “an $#lächeln $el/V(sep)$” which is where the separable verbs are. The classification of the relevant properties of the verb have been indicated in the GLEX form, and therefore the GLEX entry may result in several entries in the TWOL lexicon (which is a technical matter of compiling the source description into some operational form).

\[
\begin{align*}
\text{LEXICON VERB} & \text{ an SEPCONT;} \\
\text{LEXICON SEPCONT} & \# \text{ VSEPBASE; \\
\text{LEXICON VSEPBASE} & \text{ läch el/V(sep);} \\
\text{LEXICON el/V(sep)} & \text{ ele2n: V1el/V(sep);} \\
\text{LEXICON V1el/V(sep)} & \text{ +V+INF:ele2n, $#; \\
\text{LEXICON V1el/V(sep)} & \text{ +V+IND+PRÄS+SG1:le \#; \\
\text{LEXICON V1el/V(sep)} & \text{ +V+KONJ+PRÄS+SG1:le \#; \\
\text{LEXICON V1el/V(sep)} & \text{ +V+KONJ+PRÄS+SG3:le \#; \\
\end{align*}
\]

The setup for the correspondence for “anlächeln+V+KONJ+PRÄS+SG3” — “anlächle” is marked with “!~**” above, and is as follows:

\[
\begin{align*}
\text{Base} & \text{ Lexical} & \text{Surface} & \text{Lexicon of entry} \\
\text{a} & \text{a} & \text{a} & \text{VERB} \\
\text{n} & \text{n} & \text{n} & \text{VERB} \\
\# & \# & \# & \text{SEPCONT} \\
\text{l} & \text{l} & \text{l} & \text{SEPVBASE} \\
\text{a} & \text{a} & \text{a} & \text{SEPVBASE} \\
\text{c} & \text{c} & \text{c} & \text{SEPVBASE} \\
\text{h} & \text{h} & \text{h} & \text{SEPVBASE} \\
\text{e} & \text{e} & \text{e} & \text{el/V(sep)} \\
\text{l} & \text{l} & \text{l} & \text{el/V(sep)} \\
\text{c2} & \text{c2} & \text{c2} & \text{el/V(sep)} \\
\text{n} & \text{n} & \text{n} & \text{el/V(sep)} \\
\text{+V} & \text{1} & \text{1} & \text{V1el/V(sep)} \\
\text{+KONJ} & \text{e} & \text{e} & \text{V1el/V(sep)} \\
\text{+PRÄS} & \text{0} & \text{0} & \text{V1el/V(sep)} \\
\text{+SG3} & \text{0} & \text{0} & \text{V1el/V(sep)} \\
\end{align*}
\]

Again, the two-level rules relate the second and the third column to each other. The actual base form and analysis shown to the user (“lächeln+V+KONJ+PRÄS+SG3”) is generated with the same rules but in a reverse direction.

The morphophoneme “e2” in the base form has been established and accompanied with two-level rules because the presence or omission of the “e” in infinitives (and other similar endings) can be stated in terms of the phonological context. (The present example word alone does not, of course, justify these decisions.)

\[
L = l \ r ;
\]

Default correspondence: e2:e

\[
\begin{align*}
\text{“e2.0”} & \Rightarrow [C: \ x ; i | [a | \text{e} | \text{e} | \text{e} : a] \% : e : [L: | r | I] ; \\
\text{“e2.0 opt”} & \Rightarrow V \% : e : h ^ \text{”} - n ; \\
\text{“e2:e”} & \Rightarrow e2:e \Rightarrow C : i \% : - \text{”} - ;
\end{align*}
\]

The first rule accounts for infinitives like “handeln0n”, the second the optional omission of “e”, and the third one forbids the omission in stems ending in single vowel “i” (“kni-en”).

### 1.3 Transparency and linguistic motivation of the rules

Word forms “Tisch”, “Tisches”, “Tischen” get the following analyses.

\[
\begin{align*}
\text{Tisch} & \text{ Tisch+S+MASK+SG+DAT} \\
\text{Tisches} & \text{ Tisch+S+MASK+SG+AKK} \\
\text{Tischen} & \text{ Tisch+S+MASK+SG+NUM} \\
\end{align*}
\]

The entry for “Tisch” comes from the

*LDV–Forum Bd.11, Nr.1, Jg.1994*
CGD and has the following format in the GLEX file:

\[ m \text{ tisch } S1(s/es) \]

This entry is converted trivially into a TWOL entry, and positioned in the SUBST lexicon:

LEXICON Root
SUBST;
...
LEXICON SUBST
tisch S1(s/es)/m;
...
LEXICON S1(s/es)/m
S1(s/es)/m/end;
...
LEXICON S1(s/es)/m/end
Sg1s/es/m/end;
P1l/m/end;
LEXICON Sg1s/es/m/end
+S+MASK+SG+GENs: ##;... 
Note that even rare (SELTen) forms have been accounted for. These entries can be eliminated if needed automatically. There is a number of distinct series of endings for singular and another set of series for plural. There would be many more distinct series if both were given at the same time.

The next examples “vorbeischwammst”, “vorbeischwämme”, “vorbeigeschwommen” exemplify a few points. One is the productive combination of the prefix and the verb, and the other is the need to describe the idiosyncratic behavior of “schwommen” just once in the description. All irregular (or strong) verbs have been coded once in a tabular form where they are classified (e.g., U2) and all its stems are given:

\[ S6L\text{ENT.V5Ua schwomm ab- an-}
davon- durch- hinaus- hinüber-
ver.* herüber- ab- an- durch.* frei-

Note that the table also includes a still more idiosyncratic form “schwämm” as an alternative to “schwömm”. The table also contains a list of prefixes applicable to this verb along with the coding whether the result is inseparable (:), and whether the past participle is formed without prefix “ge” (*).

Here are the analyses of the example words:\(^2\)

vorbeischwammst
vorbeischwämme
vorbeigeschwommen

LEXICON VERB
vorbei SEPCONT; ! Not automatically from CGD
...
LEXICON SEPCONT
# VSEPBASE;
...
LEXICON VSEPBASE
schwämmen+SG schwömme V4U;
schwämmen+SEL TEn+SG schwömme V5Ua;
... 
LEXICON VSEPBASE-GE
schwämmen+SG schwömme V6U(sep);
... 
LEXICON V4U
+V+IND+PRÄT+SG2=1t ##;... 
LEXICON V6Ua
+V+KONJ+PRÄT+SG1=3e ##;... 
LEXICON PARTEND
vorbei#ge#schwamm V6U(sep)/adj;
...
LEXICON V6U(sep)/adj
A01s-en-part; LEXICON A01st-en-part
en+A(PART)+POS en ADJ;

\(^2\)Note that the third example word here is a wrong one. The cited entries cover the correct “vorbeigeschwommenen” equally well.
**LEXICON ADJ**
+SG+AKK+MASK:en
+SG+GEN+MASK:en
+SG+DAT+MASK:en
+SG+GEN+NEUTR:en
+SG+DAT+NEUTR:en
+SG+GEN+FEM:en
+PL+NOM:en
+PL+AKK:en
+PL+DAT:en
+PL+GEN:en

Default correspondence: e4:e

```
"e4,e" ↔ e %:n "":
```

The next examples “Hausdächern”, “Häusermeers” exemplify compounding and this is the area where significant effort has been spent in designing GERTWOL, because this (rather than derivation) is decisive for the coverage and usability of the system. There is no way of listing enough compounds as ready made entries in a dictionary because compounds are made on the fly when needed.

The following are analyses using the TWOL engine. There is a word boundary (\textasciitilde) between the components which is shown by the program if so desired.

\texttt{Hausdächern}
\begin{verbatim}
Haus\#dach+S+NEUTR+PL+DAT
\end{verbatim}

\texttt{Häusermeers}
\begin{verbatim}
Häuser\#meer+S+NEUTR+SG+GEN
\end{verbatim}

The entries for “Haus", “Dach" and “Meer" in the GLEX file have been classified (automatically) as nouns participating the normal compounding, and therefore the TWOL entries were placed in the SUBST lexicon. For a great number of nouns, the glue element after the stem can be determined based on the inflectional classification, or the shape of the stem. For some others, there are more alternatives open. Some of these could be excluded, but the effort needed to carry it out for the whole vocabulary is substantial.

**LEXICON SUBST**
haus S4+(es)/nt;
dach S4+(s/es)/nt;
meer S1(s/es)/nt;
...

**LEXICON SCONT**
# SUBST;
...!

**LEXICON S4+(es)/nt
SCONT;
es SCONT;
erSCONT;
...!

**LEXICON S4+(s/es)/nt
S4+(s/es)/end;
...!

**LEXICON
S4+(s/es)/nt
P4+/nt/end;
...!

**LEXICON S1(s/es)/nt
Sgl/es/en/end;
...!

**LEXICON Sgl/es/nt/end
+S+NEUTR+SG+GENS
...!

**LEXICON P4+/nt/end
+S+NEUTR+PL+DAT+ern@U
...!

The following example “Unabhängigkeitserklärung” refers to compounding and derivation. The GLEX contains:

\texttt{f unabhängigkeit S9en}

The former comes from CGD as a ready made derivation (probably we would like to make “-keit” into a more productive derivational suffix, but hardly any need has shown up yet). All nouns ending in “-eit” are feminine, and have a fixed and obligatory compounding glue element “s” (no glue and “en” are forbidden).

\texttt{V01* erkl->-eit}<r i

This entry for the verb is converted into both as a verbal entry for “erklären” and as a derived noun “Erklärung”. Distinct entries are produced automatically, and separate entries are needed because the compounding characteristics of verbs and nouns are quite different.

Here, again, is the analysis:

**Unabhängigkeitserklärung**
Unabhängigkeitserklärung+S+FEM+SG+GEN
Unabhängigkeitserklärung+S+FEM+SG+DAT
Unabhängigkeitserklärung+S+FEM+SG+AKK
Unabhängigkeitserklärung+S+FEM+SG+NOM

The lexicon entries are inserted in the expected place, SUBST lexicon. Two entries are generated for “unabhängigkeit”, one for the inflectional paradigm and the other for

\textit{LDV–Forum Bd.11, Nr.1, Jg 1994}
the selective compound glue element. Because compounding behavior ("-s-" ) is relatively independent of the inflectional class (S9en/f/end), it is simpler to use separate entries. (There is no penalty for any of the implementations in terms of space or speed.)

LEXICON SUBST
unabhängigkeit S9en/f/end;
unabhängigkeit SCONT-s;
erklä.rung s-ung;
...
LEXICON SCONT
# SUBST;
...
LEXICON SCONT-s
s SCONT;
LEXICON S-ung
S9en/f/end;
...
LEXICON S9en/f/end
Sg3/f/end;
...
LEXICON Sg3/f/end
+S+FEM+SG+ NOM. #;#
+S+FEM+SG+AKK: #;#
+S+FEM+SG+DAT: #;#
+S+FEM+SG+GEN: #;#

The following is the example word "un...
lesbares". Presently "-bar" derivation is not handled in a productive manner because many of the entries of this kind are already given in CGD. We have doubts whether this derivation is possible for all verbs.  

LEXICON
ADJ-NONCOMP
un ACONT1;
| un- permitted only
| at the beginning of a
| word, thus here

LEXICON ACONT1
| ASTART;
| is a boundary
| (weaker than a word
| boundary)

LEXICON ASTART
le_sba_r A01r;
! From CGD

LEXICON A01r
+A+POSI ADJ;

LEXICON ADJ
+SG+ NOM+ NEUTR: e4s
#;#
+SG+ AKK+ NEUTR: e4s
#;#;

The following is the example word

durchdachte" where the underlying "denken" is a weak irregular verb, or "dachen" which is a regular verb. Both verbs may follow the initial component "durch". There are two kinds of "durch": one separable and the other inseparable.

durchdacht

durchdacht+A(PART)+POS+SG+AKK+FEM ...

durchdenken+V+IND+PRDT+SG1

durchdenken+V+IND+PRDT+SG3

durch#denken+V+IND+PRDT+SG1

durch#denken+V+IND+PRDT+SG3

durchdachen+V+IND+PRDT+SG1

durchdachen+V+IND+PRDT+SG3

durch#dachen+V+IND+PRDT+SG1

durch#dachen+V+IND+PRDT+SG3

durchdachen+V+KONJ+PRDT+SG1

durchdachen+V+KONJ+PRDT+SG3

durch#dachen+V+KONJ+PRDT+SG1

durch#dachen+V+KONJ+PRDT+SG3

(and 7 incorrect forms with the A(PART)
with the boundary ...oops!)

LEXICON VERB

LEXICON SEPCONT
SSEPBASE;

LEXICON INSEP
SSEPBASE;

LEXICON VSEPBASE
d0000enken:dachte V4R;
dach V01(sep);
! 'roof' V

LEXICON VSEPBASE
d0000enken:dachte V4R;

dach V01*

LEXICON V01(sep)
e2eite V4R;
e2eite V5R;

LEXICON V01*
e2eite V4R;
e2eite V5R;

LEXICON V4R
+V+IND+PRÄT+SG1:
#;#
+V+IND+PRÄT+SG2:
#;#

The following is the example word

LDV-Forum Bd.11, Nr.1, Jg.1994
LEXICON V5R
+V+KONJ+PRÄT+SG1:
###;
+V+KONJ+PRÄT+SG2:
###;

LEXICON PARTEND
durchdocht V6R/adj;
...
LEXICON V6R/adj
e1t A01est/st-part;
LEXICON V6R(sep)/adj
e1t A01est/st-part;
LEXICON A01est/st-part
+A(PART)+POS: ADJ;
! Adjectivized participle (inflected)

LEXICON ADJ
+SG+NOM+FEM:e4
###;
+SG+AKK+FEM:e4
###;
+SG+NOM+MASK:e4
###;
+SG+NOM+NEUTR:e4
###;
+SG+AKK+NEUTR:e4
###;
+PL+NOM:e4
###;
+PL+AKK:e4
###;

Cl = b c d f g j k p q s t v w x s l t l b;

Default correspondence: e1:0
"e1:0"
  = [C1 : C b h [m | n] . ;
     [t | d . ;
     C i % " . ;

Rules for "e2", "e4" as before.3

The next examples "gut", "besser", "besten" show how suppletion is typically handled. Instead of one entry, there will be three all give "gut" as their base form. Note that all remaining characteristics of inflection and compounding can be shared by placing these entries in an appropriate lexicon and assigning a suitable continuation class to them. (The zeroes (0) in the lexical entries are there for technical reasons only, in order to keep the character by character correspondences simple. The zeroes are mostly added automatically when entries are generated, and the reader may well ignore their presence.)

LEXICON ADJ-NONCOMP
gut+ADJ+POS: gut ADJ;
...

LEXICON ASTART
00000gut+A+KOMP:besser ADJ;
000gut+A+SUP:bes00t ADJ;

000gut00+A+SUP2:bes00t1en ###;
...
LEXICON ADJ
###;
+SG+AKK+MASK:e4n ###;
+SG+DAT+MASK:e4n ###;
+SG+GEN+MASK:e4n ###;
+SG+DAT+FEM:e4n ###;
+SG+GEN+FEM:e4n ###;
+SG+DAT+NEUTR:e4n ###;
+SG+GEN+NEUTR:e4n ###;
+PL+NOM:e4n ###;
+PL+AKK:e4n ###;
+PL+DAT:e4n ###;
+PL+GEN:e4n ###;
...

1.4 Morpho-syntactic analysis (categories)

There is a separate paper which describes the set of morphosyntactic tags or features used in GERTWOL. Where word-forms are homographic, multiple analyses are produced, one for each interpretation. This is intended for the convenience of syntactic processing which might follow the morphological analysis. If only the base form and its part of speech is desired, then the extra features can be deleted, and the user gets only one base form plus part of speech.

Prepositions have valency information (+Dat, +Acc, +Gen), and positional information (+pre, +post). Pronouns have an indication of their possible positions in an NP, i.e., attributive or independent, (+Det) indicates the possibility of an attributive position which is the normal case for some, and unusual for others.

The syntactic significance of capitalization is preserved, e.g., "Arbeiten" analyzed as "arbeiten + V IND PRAS PL3" would be possible only at the beginning of a sentence.

1.5 The handling of generation

The Xerox Lexical Tools includes a program called INFL which is capable of both analyzing word-forms and generating them. This is done by the same program from the same data structure, just by switching the mode. The input for the generation is the same as what the analysis gives as output.

LDV–Forum Bd.11, Nr.1, Jg.1994
There is also a “prefix” mode where one may drop some of the features at the end, and the INFL program then generates with all possibilities, i.e. the part of the paradigm, or the full paradigm.

The original TWOL engine is not sensitive to what direction it is being used in. Normally the dictionary is for analysis. With a simple utility program the fields can be switched so that the resulting dictionary will do the generation. Thus the same description is underlying both modes, and so is the same runtime program, but the input file is formatted in a different way.

Bidirectionality has always been one of the fundamental principles of the two-level morphology. Indeed, the normal mode of analysis in TWOL uses the generation mechanism to build the correct base form on the fly (especially when there is no ready-made entry for a derived entry).

1.6 Applications to other languages
The two-level model had been applied to at least 30 languages. The following have a full-scale lexicon and are capable for handling running texts (lexicons with 40-100,000 entries):

- Finnish, Swedish, English,
- Russian, Swahili, Estonian, Danish, Basque, French, Arabic, Lapplish.

There are many others with fairly comprehensive rules and model words for all inflectional types and a few thousand word dictionary (Latin, Mari, Assyrian, Babylonian, Savo dialect of Finnish). Many more less comprehensive descriptions exist, e.g., for Sanskrit, Nenets, Polish, Turkish, etc.

In principle, most types of languages have been encountered. English can be done without actual two-level rules at all (as the one which is part of ENCG), or with more extensive rules (the version by Lauri Karittunen at PARC). Finnish is an example of a language where morphophonemes prove to be useful in simplifying the description, because the processes are regular and general. Also some beautiful linguistic generalization may be achieved in the two-level framework which are not easy to express in the rewriting framework. The same comments apply to Sanskrit as well, but the magnitude of overall complexity is far greater there, as the linguists know.

2 Technical design and practical use

2.1 Conceptual goal of the design
GERTWOL was designed to be an industry strength, efficient, wide coverage, general purpose and accurate analyzer/generator which could be used in a wide variety of applications.

Much care was taken to make the basic description as solid as possible so that there would be little need to return to it later for tuning or revision. Thus, the basic inflection is very carefully designed and checked. The lexicon has also been tested on large amounts of corpus text for validation (in cooperation with the University of Stuttgart, some 30 million words of text).

The vocabulary is comprehensive from the very beginning through the use of Collins German Dictionary data as the main source (through an arrangement with HarperCollins). In this way, the need to add more entries is minimized. In spite of this, easy, and even fully automatic inclusion of new entries is provided for.

Efficiency is of primary importance as many application assume the processing of gigabytes (and some terabytes) of text data. The TWOL program can achieve a very satisfactory speed, and the Xerox Lexical Tools allow for speeds of about 250 GB per hour with a highly compressed dictionary.

Hardly any attention has been paid now to the setup where a lexicographer is describing a new language, simply because German morphology is already extremely well accounted for, and there exist very high quality dictionaries. We simply could not imagine of repeating that effort. We would not have the resources to accomplish such a task (of tens or hundreds of man years). Furthermore, German is not a very suitable language for making a theoretically and educationally valuable description, because there is fairly little morphophonol-
ology which would be regular and productive. Most of the complexity lies in the combinatorial part, and that is determined much by convention rather than regular principles.

2.2 Portability of software and data
The TWOL program was originally written in Pascal and later converted into C. The Pascal version was run on Burroughs B6700 and on MSDOS machines. The C version has been ported to a wider set of machines and platforms, at least to: MSDOS machines, Macintosh, Sony NEWS Unix BSD 4.3, IBM RT AIX Unix V, Sun 386i SunOS Unix, Sun 3 SunOS Unix, Sun SPARC-station (various models) SunOS 4.1.X and Solaris 2, HP 9000 Unix V, SCO Unix V, Digital VAX VMS, IBM mainframes MVS, Apollo Domain Unix.

There is also a Prolog version of the TWOL which works on SICSTUS Prolog. It uses a lexicon and a rule component mechanically converted from the C TWOL version into a Prolog clause.

The Xerox Lexical tools can accommodate the GERTWOL lexicon and rules after a mechanical conversion of the format. Those tools and runtimes exist at least on Macintosh and Sun SPARCstations at present.

The lexicons in binary format are portable between various platforms with the limitation of the byte order differences (e.g., SPARC vs. Intel processors), and the need to handle character code conventions (ISO Latin1, PC code, Macintosh code) with due care. Programs are in no way dependent of any particular character coding.

2.3 Interface to syntax and semantics
GERTWOL is intended to be used in various contexts such as processing of large text corpora, information retrieval, spelling checking, and as the input module for various syntactic or semantic parsers. As a principle, it seems appropriate when designing morphological analyzers for languages with complex morphology to restrict the task of the morphological analyzer to the basic morphology.

The output of the basic morphological analyzer should be used as a key to syntactic and semantic lexicons with richer and theory specific coding. Such postprocessing of the morphological analysis should perhaps occur starting from output where the words and derivational elements are canonized. In this way it would be straightforward to describe the subcategorization features (or case the frame) of derived words and compounds correctly.

For languages with a more trivial morphology, like English, we have chosen to include surface-syntactic features in the lexicon along with the plain morphological tags.

2.4 Aiding the user
Debugging is typically needed for the rule component, inflectional paradigms of model words in each inflectional class, and to some extent, in controlling the continuation patterns in the lexicon.

The rule compiler TWOLC has extensive facilities for verifying the rules, and for tracing the actual behavior of existing rules. One can instantly try what kind of surface forms will be generated:

```
twolc> lex-test
Lexical string ("q" = quit):
 l
 e
 r
 n
 e1:0
 t
e
```

In case the expected surface forms do not show up, or to determine which rules are responsible for blocking certain ungrammatical realizations, one can use “pair-test”:

```
twolc> pair-test
Lexical string ("q" = quit):
 l
 e
 r
 n
 e1:e
REJECTED: "e1:e" fails in state 2.
```

The LEXC lexicon compiler has a very useful feature for checking the model words by generating full paradigms of forms for each base form. These are checked manually in order to detect possible missing or extraneous, or incorrectly formed forms.
Once the paradigm has been validated, the verification can be done automatically using the Unix 'diff' program for fast validation after changes to the ending lexicons or rules.

The continuation mechanism of the dictionary is fairly complicated to follow in the actual dictionary files. Therefore, there are specific tools (twol-tree) for viewing the possible sequences of morphemes, e.g.:

```
$ twol-tree -twol -c 3 ger.complete.dic | more
TwolRoot
 | Root
 | # [5610]
 | INTJ [297]
 | | # [2] ...
 | 1KORJ
 | uKORJ [63]
 | SUBST [3]
 | SUBST-RTPS [2]
 | SUBST-RCAP [2]
 | | SCOWTH-RCAP
 | | DIGIT [30]
 | | | DIGIT1
 | | | | DIGIT2
 | | | | | SCOWTH-RCAP [5] ...
 | | S2(-)/at/end [5]
 | | Sgl/-int/end
 | | P12/at/end
 | | SCOWTH-RCAP [445]
 | | SUBST-RCAP ...
 | | SUBSTPREF-START...
```

The TWOL engine has some switches for debugging, e.g. the following one shows the path along which the analysis proceeds:

```
$ tw-ger -Tging
"<ging.>"
  → #
  → Root
ging → V5U
ging → V4U
ging → #
ging → #
g → #
g → #
  "geh" V IND PRÄT SG3
  "geh" V IND PRÄT SG1
```

2.5 Limits to the size of the system
On the whole, there are hardly any limits set by the TWOL program itself. The availability of core memory in the target computer is the actual constraint the size of the lexicon, and this constraint has ever less significance. The runtime TWOL program allocates memory space according to the actual size of the saved lexicon. There is some, high enough limit for input line and input word length (say 1-10 kB) which can be adjusted as needed. Excessively long inputs will be discarded.

The development versions of TWOL programs have parameters according to which one can adjust the upper limits in those areas where longer areas are needed through command line switches.

The versions made using the Xerox lexical tools need (in most cases) less memory than the TWOL.

2.6 Interface to non-ASCII characters
The details which character codes will be used can be adjusted according to the need, and we are prepared to handle three options for German:

8 bit ISO Latin1 ä ö ü Ä Ö Üß
8 bit ISO Latin1 ä ö ü Ä Ö Ü ß (as often in Switzerland)
7 bit ASCII ae ou ue Ae Os Ue ss

E.g., to use "ß" instead of "β" is a matter of excluding the rule which is responsible for this convention.

2.7 User friendliness of the ‘turn around’
The vocabulary is extensive and additions are not required often. When large amounts of text are being processed e.g., for surface syntactic analysis, we have planned to use a separate module which we call “morphological heuristics”. This would give each unanalyzed word a set of possible base-forms and morphosyntactic tags. Experience from English suggest that almost all instances can be handled in this way.

In some applications, it is desirable to add new words to the GERTWOL lexicon on the fly. This is common in certain types of information retrieval applications, and it is handled by coupling an entry generator (EGEN) with the Twol program. During the execution, the input may consist either of word-forms to be analyzed, or new word entries which are immediately installed into
the lexicon. New entries may have either the form of a normal GERTWOL entry such as:

*ghotbzadeh NAME-M/F;

In an actual application, the user would add only a brief part of speech classification in front of the base form of the new word, e.g.:

NN: Ghotbzadeh

This entry is converted automatically by the entry generator into the previous format, which is then instantly added to the dictionary. (This facility has been used for about two years in a newspaper archive application of the largest Finnish newspaper publisher for the Finnish texts. By chance, this facility is not in the latest version of TWOL, but it will be reactivated in the near future. In the meantime, one can use the stand-alone entry generator, and cut and paste the resulting entries.)

New entries can be added by editing the master files, if one so wishes. This is typically done by those who maintain the dictionary. The update cycle of rebuilding the whole GERTWOL from its GLEX source and various components might take up to a few hours, but the time depends on how extensive modifications were made. Converting the German TWOL lexicon and rule automata file into a binary file used by the runtime TWOL program takes about 15 minutes.

2.8 The state of the documentation


The interface and use of LingoSoft version of the Twol program is documented in a document by LingoSoft, and a few Unix style man pages.

The overall design and description of the GERTWOL and the system of morphosyntactic tags is documented in Mariikka Haapalainen’s forthcoming paper, out of which an abbreviated version is attached.

The lexical content from the Collins German Dictionary is, of course, documented in the published dictionary. Other words, such as proper nouns, have been added, but that list is not available.

2.9 Availability and maintenance

The GERTWOL analyzer will be made available for the academic research community at a nominal price which is intended to cover the administration and distribution costs. The academic licenses will permit normal and reasonable research and educational use.

Simultaneously, the product will be available for commercial use at a different type of license where there is a fee or royalty according to the intended use. Queries should be directed to LingoSoft, Inc., Manager, International sales, Mr. Eugene Young, Museokatu 18 A 3, FIN-00100 Helsinki, Finland. There is a separate leaflet describing those aspects.

3 The test data

The GERTWOL has been tested against word forms in a 30 million word corpus. The results were:

99% of all correctly spelled word-forms in the corpus
98% of all words in the corpus

LDV-Forum Bd.11, Nr.1, Jg.1994
4  The performance figures on previously known test data

Coverage:  
15667 input tokens total (incl. punctuation)  
15415 of these tokens got an analysis  
252 tokens didn’t get an analysis

Speed on a Sun SPARCstation 2 (without preprocessing) with the TWOL engine:

216 words per sec full analysis  
994 words per sec if only recognition of correctness is performed

Speed on a HP (without preprocessing) with the TWOL engine:

398 words per sec full analysis  
1978 words per sec if only recognition of correctness is performed

Space needed with the TWOL:

41 kB RAM for the TWOL runtime program  
6.2 MB RAM for the lexicon and rules

5  Portability

On a Intel 80486 based PC the performance figures are similar to those on a SPARCstation 2. On an older 80386 based PC about three times slower than on SPARCstation 2. The TWOL engine takes about the same space on each platform.

Using the Xerox Lexical Tools the space requirement can be reduced into 500 kB to 1 MB. The speeds are nevertheless much higher: at least 2000 words per second for analysis, and some 10,000 words per second if only recognition is needed.

---

4There were problems in porting the preprocessing programs to the HP machines in Erlangen. The faster “flex” version did not work at all because of some unidentified problem (in contrast to the SUN and HP machines in Helsinki). Another, slower Perl version worked partly, but produced somewhat erroneous results. (Again, the error could not be reproduced on a similar machine in Helsinki.) Consequently, in both of the texts used in the benchmark, the first word of each sentence got garbled, and consequently, was rejected.

5The HP workstation “sol” was twice as fast.

6The MSDOS version was brought to Erlangen, but we were not able to move the files from the Unix machine to the PC in order to install and run it.

7A test version of the version using the Xerox Lexical Tools was present on a Macintosh Powerbook, but not ported to the official benchmark compiler, and not officially measured.