TAGH: A Complete Morphology for German based on Weighted Finite State Automata - draft

Alexander Geyken[1] and Thomas Hanneforth[2]

[1] Berlin-Brandenburg Academy of Sciences
[2] University of Potsdam

Abstract. TAGH is a system for automatic recognition of German word forms. It is based on a stem lexicon with allomorphs and a concatenative mechanism for inflection and word formation. Weighted FSA and a cost function are used in order to determine the correct segmentation of complex forms: the correct segmentation for a given compound is supposed to be the one with the least cost. TAGH is based on a large stem lexicon of almost 80.000 stems that was compiled within 5 years on the basis of large newspaper corpora and literary texts. The number of analyzable word forms is increased considerably by more than 1000 different rules for derivational and compositional word formation. The recognition rate of TAGH is more than 99% for modern newspaper text and approximately 98.5% for literary texts.

1 Introduction

Compounding in German is productive, therefore full-form lexicons cannot cover German morphology completely. Hence morphology programs such as Gertwol (Haapalainen and Majorin[6]) or canoo [`http://www.canoo.net`] generally use stem lexicons together with decomposition and derivation rules. The present approach differs from the aforementioned morphology systems in that it is not based on a two-level-morphology but on a concatenative mechanism. The formal prerequisites for that are presented in section 2, 'Morphology and Weighted Finite State Automata'. In section 3 we will describe the linguistic aspects of the TAGH-morphology, the lexicon and the word formation rules. In addition, the semantic types of LexikoNet, a shallow semantic hierarchy for German nouns will be described. All noun entries of the stem-lexicon are annotated with these semantic types. Thus, it is possible to use semantic types both for expressing word formation rules as well as for the semantic typing of semantically transparent compounds. In section 4, the problem of disambiguation of ambiguous morphological analysis is addressed. Section 5 and 6 summarize the current state of development of the presented system and sketch out some ideas for future work.

[draft] A. Yli-Jyrä et al. (Eds.): FSMNLP 2005, LNAI XXXX, pp. 1 – **??**, 2005

2 Morphology and Weighted Finite State Automata

2.1 The Morphology Problem

Basically the morphology problem can be stated as follows: given an input alphabet Σ_I, an output alphabet Σ_0 and a morphological alphabet [3] Σ_M, a morphology realizes a partial function $\Sigma_I^* \rightarrow \wp(\Sigma_0^*.\Sigma_M^*)$. The morphological alphabet consists of letters, morpheme boundary symbols (#,∼), categories (like NN, NSTEM, NSUFF), and features like 3, sg, nom.

Since a string in Σ_I^* can be mapped to several output strings we need the power set operator $\wp$. Usually this function is considered a rational one, that is, both Σ_I^* and $\Sigma_0^*.\Sigma_M^*$ are supposed to be regular languages. This decision rules out constructing word-syntactic trees and also the treatment of unrestricted reduplication phenomena in certain languages. How do we construct such morphology functions? There are at least two ways based on closure properties: one is based on the fact that regular languages are closed under substitution, the other exploits the closure of regular languages/relations under intersection/composition (e.g. Hopcroft [7]). In the next section we will sketch the algebraic specification of a weighted finite state transducer representing a morphology function for German.

2.2 Algebraic Specification of a Morphology for German

The first building blocks of the system are two lexicons, one for stems and one for affixes. Fig. 1 shows a very small fraction of the stem lexicon containing two verbs and a noun.[4] We have decided to handle irregular, nonpredictable allomorphy like *Umlaut* and *Ablaut* in the lexicon, that is, the stem lexicon is represented by a finite state transducer (Fig. 3 in appendix), which maps lemmas to allomorphic stems (cf. also Karttunen [9]).

For example, the irregular verb werfen (to throw) is mapped to its five allomorphic stems. A number of mostly boolean features like $StPret$ encodes important morphological properties like co-occurrence restrictions, in particular the fact, that a certain stem can be used only in an inflected form of a certain type. $StPret = yes$ for example means that a stem marked in that way must be used in preterite verb forms. In effect the seven boolean stem features define the stem equivalence classes for every irregular German verb. The affix lexicon which similarly contains categorized prefixes, derivational and inflectional suffixes, infixes etc. is compiled into a finite state acceptor. The next step consists

[3] We assume the morphology alphabet to be disjoint from the input and output alphabet.

[4] We use the AT&T LexTools notation, cf. Sproat [14]). Symbols in [ ] denote possibly underspecified categories. Features are defined with respect to an inheritance hierarchy and are represented as transition labels. Underspecification is realized as the disjunction of all maximal subtypes of a super type.

```
(rett:rett)  [VREG VType=main PrefVerb=no Latinate=no PartIIIrreg=no]
(werf:warf)  [VIRREG VType=main PrefVerb=no\
             Latinate=no StDef=no St23SgInd=no StPret=yes StSubjI=no\
             StSubjII=no StPartII=no StImpSg=no St23SgIndVowelChange=yes]\
(werf:werf)  [VIRREG VType=main PrefVerb=no
             Latinate=no StDef=yes St23SgInd=no StPret=no StSubjI=yes \
             StSubjII=no StPartII=no StImpSg=no St23SgIndVowelChange=yes]\
(werf:wirf)  [VIRREG VType=main PrefVerb=no Latinate=no StDef=no\
             St23SgInd=yes StPret=no StSubjI=no StSubjII=no StPartII=no\
             StImpSg=yes St23SgIndVowelChange=yes]\
(werf:worf)  [VIRREG VType=main PrefVerb=no Latinate=no StDef=no\
             St23SgInd=no StPret=no StSubjI=no StSubjII=no StPartII=yes\
             StImpSg=no St23SgIndVowelChange=yes]\
(werf:w\"urf)[VIRREG VType=main PrefVerb=no Latinate=no StDef=no \
             St23SgInd=no StPret=no StSubjI=no StSubjII=yes StPartII=no\
             StImpSg=no St23SgIndVowelChange=yes]\
(Haus:Haus)  [NSTEM Gender=neut NICSg=ic_sg1 NICPl=ic_pl3\
             StemType=deko Bound=no DecoActive=yes]\
(Haus:H\"aus)[NSTEM Gender=neut NICSg=ic_sg1 NICPl=ic_pl3\
             StemType=deko Bound=no DecoActive=yes]
```

Fig. 1. extract of the stem lexicon

in taking the union of the lexicons and afterwards the closure of this union :

$$Morph^* =_{def} (Stems \cup ID(Affixes) \cup ID(MorphBoundaries))^* \ [5]$$

The regular relation $Morph^*$ denotes the infinite language of all sequences of stems and affixes and their features without taking into account word-grammatical restrictions or phenomena of regular allomorphic variation when certain morphemes are concatenated. To solve the first problem, $Morph^*$ is composed with a word grammar:

$$Morph_{WG} =_{def} Morph^* \circ ID(WordGrammar).$$

This filters out all sequences which are ill-formed according to the word grammar. $WordGrammar$ must be defined as a regular set. Fig. 2 shows some example rules given as regular expressions which are disjunctively combined (of course, the actual grammar is defined in a much more modular fashion):

```
(([Letter]*) [NSTEM] #)* ([Letter]*)  [VREG] ~ ung [NSUFF Gen=fem][NINFL Num=sg Case=*]
(([Letter]*) [NSTEM] #)* ([Letter]*)  [VREG] ~ ung [NSUFF Gen=fem] en [NINFL Num=pl Case=*]
(([Letter]*) [ASTEM] #)* ([Letter]*)  [VREG] ~ bar [ASUFF]
(([Letter]*) [NSTEM] #)* ([Letter]*)  [VREG] ~ bar [ASUFF] ~ keit [NSUFF Gen=fem]\
             [NINFL Num=sg Case=*]
```

Fig. 2. sample regular word grammar

Rules 1 and 2 for example describe the nominalization of regular verbs by means of the suffix -ung: retten (to rescue) $\rightarrow$ $rett \sim ung$. Rule 3 defines the

[5] ID(A) represents the identity relation of a regular set A; $MorphBoundaries$ is the set of morpheme boundary symbols: $\sim$ for suffixation, # for compounding etc.

suffixation of verb stems with the suffix -bar: retten $\vdash$ rettbar (rescuable), while rule 4 allows a further suffixation of -bar-suffixed verbs with the suffix -keit, resulting in forms like rett∼bar∼keit (rescuability). All four rules permit an unlimited number of noun stems which precede the derived verb stems, resulting in (rather senseless) compounds like Haus#rett∼bar∼keit (house-rescuability). Fig. 4 (appendix) shows the FSA associated with the word grammar.

The next step consists of applying morphographematic rules, so-called spelling rules, to the outcome of $Morph^* \circ WordGrammar$ to handle regular allomorphy like *schwa*-insertion etc. These types of rules are defined by context-sensitive replacement rules, which can be modelled as finite-state transducers with the restriction that they are not applied to its own output, (see Kaplan [8]). The following rule describes the insertion of *schwa* after verb stems ending with *tt* before a set of the verbal inflectional elements:

$\epsilon \rightarrow$ e / $tt[VREG]$ _ (n(d?) $|t|st|t(e|est|en|et))$ [VINFL]

This accounts for word forms like rettest *(2. sg pres)* or rettetest *(2. sg pret)*. All k spelling rules SRi are composed into a single spelling transducer (of course the ordering of these rules is of importance):

$Spelling =_{def} SR1 \circ SR2 \circ \cdots \circ SRk$

After composing $Morph_{WG}$ with $Spelling$ we obtain a transducer representing a relation between lexical forms and surface strings, both interleaved with categorical information. To define the input tape of the morphological analyzer we have to delete the symbols of the morphology alphabet and to invert the resulting transducer:

$MorphAnalyser' =_{def} Invert(Morph^* \circ ID(WordGrammar) \circ Spelling \circ Cap \circ InputBand)$.

$InputBand$ is the composition of a sequence of rules like the following:

$([NSTEM]||[VREG]||[VIRREG]||[NSUFF]|| \sim) \rightarrow \epsilon$.

Cap ensures the correct capitalization of the surface word: sequences which define nouns start with an uppercase letter, noun stems inside of words start with a lowercase letter etc. The remaining task is to format the output band of the analyzer accordingly:

$OutputBand =_{def} RHHR \circ MorphFeatToSynFeat$.

$RHHR$ represents the right-hand-head-rule which says that in German the stem or suffix morpheme standing at the right periphery determines the morphosyntactic properties of the whole word. $RHHR$ is represented by a sequence of contextual replacement rules like the following:

$[NSTEM] \rightarrow \epsilon$ / _ $(\#| \sim)$.

This rule deletes noun stem markers preceding morpheme boundaries. Finally $MorphFeatToSynFeat$ rewrites sequences of morphological features/categories to morphosyntactic categories (like NN), followed by morphosyntactic features (case, gender, number, etc.) If, for example, the word Rettungen is analyzed,

both categories $NSUFF$ and $NINFL$ contribute to the features of the complete word: $NSUFF$ defines its gender and $NINFL$ its number and case. The final morphological transducer is defined by:

$MorphAnalyser =_{def} MorphAnalyser' \circ OutputBand.$

As usual, applying the morphology to a word consists in composing the input word given as an identity transducer with $MorphAnalyzer$ and taking the output band:

$Proj2(ID(input) \circ MorphAnalyzer).$

2.3 Morphological Complexity and Weighted Automata Morphology

Analyzers like the one sketched in the last section segment longer word forms in a sometimes absurd manner into sets of senseless alternatives amongst which to choose is not an easy task.[6] Therefore, it would be useful to have a notion of morphological complexity which can be integrated into the analyzer and which prefers for example compounds with fewer segments to compounds with more segments or favors lexicalized but morphologically complex compounds over non-lexicalized readings. A simple way to achieve this is to reconstruct the grammar as a weighted regular language where the weights reflecting the morphological complexity of the different word formation rules can be either chosen by hand or acquired through machine learning techniques. To put it a bit more generally: we define a weighted language L where each element in L is a pair consisting of a string $x \in \Sigma^*$ and a weight c chosen from a weight set W. A suitable algebraic structure for this task in the context of finite-state automata is a *semiring*. A structure $< W, \oplus, \otimes, \overline{0}, \overline{1} >$ is a semiring (e.g. Golan [5]), if it fulfils the following conditions:

1. $< W, \oplus, \overline{0} >$ is a commutative monoid with $\overline{0}$ as the identity element for $\oplus$.
2. $< W, \otimes, \overline{1} >$ is a monoid with $\overline{1}$ as the identity element for $\otimes$.
3. $\otimes$ distributes over $\oplus$.
4. $\overline{0}$ is an annihilator for $\otimes$: $\forall w \in W, w \otimes \overline{0} = \overline{0} \otimes w = \overline{0}$.

A weighted finite-state transducer A $= < \Sigma, \Delta, Q, q_0, F, E, \lambda, \rho >$ over a semiring W is an 8-tuple with Σ being the finite input alphabet, Δ the output alphabet, Q the finite set of states, $q_0 \in Q$ the start state, $F \subseteq Q$ the set of final states, $E \subseteq Q \times (\Sigma \cup \epsilon) \times (\Delta \cup \epsilon) \times W \times Q$ the set of edges, $\lambda \in W$ the initial weight and $\rho : F \mapsto W$ the final weight function mapping final states to elements in W. In section 4 we give some examples for an instantiation of the semiring template with $< \Re, min, +, \infty, \overline{0} >$, a so-called tropical semiring (e.g. Mohri [11]). This means that weights along an accepting path of an automaton are additively combined and among different paths accepting the same input string the path with the minimal weight is chosen.

[6] We give some examples in section 4.

3 Lexicon and Word Formation Rules

3.1 TAGH-Lexicon

Lexicons for NLP purposes can be represented as full-form lexicons (e.g. Courtois [3]) or as stem-lexicons (e.g. two-level morphology lexicons such as the above mentioned Gertwol or canoo). In the first case each lexicon entry corresponds to an inflected form together with its morphological features. Full-form lexicons are convenient for languages such as English or French where compounding is not productive. The lexicon look-up is then reduced to a pattern matching of a token with a lexicon entry. In the second case, a comparatively small lexicon is related to word formation rules in order to analyze word-forms which are not in the stem-lexicon. It is convenient to encode German as a stem-based lexicon because of its productive derivation and compounding.

Stem-based lexicons can be used in a concatenative or a non-concatenative way. In the latter case one attempts to describe non-concatenative processes such as the formation of irregular stems by an enrichment of the lexical entries with special symbols on which special rules apply (e.g. two-level-morphology). The TAGH-morphology relies on a concatenative stem-based lexicon as described in section 2. Hence, irregular lemmas generally correspond to several allomorphic stems. In the above-mentioned example in Fig. 1, the irregular verb werfen (to throw) has five different allomorphic stems: warf, wirf, werf, worf, würf. Likewise an irregular noun such as Haus (house) has two different allomorphic stems: Haus and Häus.

In the TAGH-lexicon a difference between simple stems and complex stems is made. A word form in the TAGH-lexicon is a **simple stem** if

(**A**) it cannot be analyzed by a morphophonetic-, derivation- or a composition-rule or a combination of them into two or more non empty segments in a way that at least one segment can be used autonomously;
(**B**) each true decomposition consists of at least one opaque segment. Here, transparency is understood synchronically; e.g. Augst [1];
(**C**) it is unmarked with respect to inflection.

According to that definition the lexicon of simple stems consists of word-forms that cannot be further decomposed (in the above-mentioned sense) in a transparent way into smaller lexemes. Of course, this definition depends on a large extent on the word formation rules as well as on the set of stems. Some examples shall illustrate this definition.

(1) Wand (wall), steh (to stand), grün (green)
(2) vorhersehbar (predictable), Drehtür (revolving door)
(3) Waldmeister (woodruff)
(4) unflätig (bawdy), drollig (funny),
(5) lexikalisch, (lexicalized), marokkanisch (moroccan)

The examples in (1) are all simple forms, those in (2) are transparent compounds since they can be analyzed by word formation rules vorhersehbar $\mapsto$ vorherseh + *bar*, Drehtür $\mapsto$ Dreh + Tür. Semantically intransparent compounds such as Waldmeister in (3) are stored in the lexicon of complex stems. Even though Waldmeister is morphologically analyzable by a simple N+N compound rule, the complex stem is the preferred lemma (cf. section 2.3 how the preference is computed, and section 4 for some examples).

The examples in (4) are encoded as simple stems since they are consistent with condition **B** of the definition above: here *un-* and *-ig* are active prefixes resp. suffixes, but the remaining word forms Flat (Old High German sauber, clean) and Droll (lower German Knirps (manikin) have no synchronous interpretation.

The examples in (5) are also considered simple stems since the current TAGH-morphology does not consider the following morphophonemic word formation rules: *-al-* in lexik-al-isch or *-an-* in marokk-an-isch.

A word form is a **complex stem** if

(**A**) it consists of at least two simple stems plus additional affixes or linking elements;
(**B**) if the meaning of the word form is either morphologically or semantically opaque, meaning that fully transparent compounds are not stored in the stem-lexicon;
(**C**) it is unmarked with respect to inflection.

Organization of the TAGH-lexicon The TAGH-lexicon itself is divided up into several sub-lexicons according to their lexical categories. There are lexicons for nouns, verbs, adjectives, adverbs, closed classes (prepositions, determiners, conjunctions), confixes, abbreviations and acronyms, each stored in a relational database (cf. table 1). Additionally, large lists of proper nouns (first names and family names as well as geographical names) were compiled.

Table 1. number of stems in the TAGH-lexicon

stem type	number
nouns	41,000
verbs	21,000
adjectives	11,000
adverbs	2,300
closed forms	1,500
abbrev, acronyms	15,000
confixes	105
family names	150,000
first names	20,000
geogr. names	60,000

3.2 Shallow Semantic Typing

All noun entries are categorized on the basis of a shallow semantic typing which has been derived from LexikoNet, a lexical ontology of German nouns (Geyken and Schrader [4]). LexikoNet is based on a concept hierarchy of more than 1,200 concept nodes that is ordered in a top-down hierarchy beginning with the concepts of CONCRETE NOUNS and ABSTRACT NOUNS.

For its use in the TAGH-morphology the LexikoNet is simplified in order to be tractable for fast analysis. The 1,200 categories are mapped to a shallow hierarchy of types selected for their prevalence in context patterns: the Brandeis Shallow Ontology, (BSO), Pustejovsky ([12]). BSO consists of the following (provisional) types: EVENT, ACTION, SPEECHACT, ACTIVITY, PROCESS, STATE, ENTITY, PHYSICALOBJECT, ARTIFACT, MACHINE, VEHICLE, HARDWARE, MEDIUM, GARMENT, DRUG, SUBSTANCE, VAPOR, ANIMATE, BIRD, HORSE, PERSON, HUMAN GROUP, PLANT, PLANTPART, BODY, BODYPART, INSTITUTION, LOCATION, DWELLING, ACCOMMODATION, ENERGY, ABSTRACT, ATTITUDE, EMOTION, RESPONSIBILITY, PRIVILEGE, RULE, INFORMATION, DOCUMENT, FILM, PROGRAM, SOFTWARE, WORD, LANGUAGE, CONCEPT, PROPERTY, VISIBLEFEATURE, COLOR, SHAPE, TIMEPERIOD, HOLIDAY, COURSE OF STUDY, COST, ASSET, ROUTE.

Shallow semantic typing is used for two purposes: for specifying word formation rules (see section 4) as well as for determining the semantic type of a transparent compound. For example, the compound noun Sprachexperte (language expert) is not part of the stem-lexicon but can be morphologically and semantically analyzed on the basis of its components (note also, that the allomorphic stem Sprach (language) is correctly lemmatized to Sprache):

```
Sprache/N#Experte[NN SemClass=Human Gender=masc Number=sgCase=nom]
```

3.3 Word Formation Rules

Approximately 1000 word formation rules are used in the TAGH-morphology in order to recognize new words on the basis of the stem-lexicon. Within the framework of the formalism it is possible to express derivation, conversion and composition rules. These rules generally operate on lexical categories, but it is also possible to restrict the applicability of a rule to subsets of lexical categories that are determined by additional features such as accented suffixes, latinate, compounding activity or semantic type. Additionally, for nouns and adjectives more than 120 non autonomous prefixes and suffixes were collected, each of them being active in word formation. It is beyond the scope of this paper to present all rules. The following examples illustrate how the encoding in the lexicon and the word formation rules interact.

(1) VSTEM + *-bar* $\mapsto$ ASTEM
(2) confix + NSTEM $\mapsto$ NSTEM
(3) ASTEM [latinate=true] + *-ist* $\mapsto$ NSTEM [semClass =Human] if ASTEM ends with *-–tär, -iv, -ell, -al.*

(4) *-chen*-derivation (only suffix) for nouns of the semantic classes ARTIFACT, PHYSICAL OBJECT or SUBSTANCE.

(1) describes the simple derivation rule: a verb stem combines with the suffix *-bar* to an adjective. Rule (2) describes a rule that combines confixes and nouns. Confixes are morphemes such as *Ergo-, Poly-, Giga-*. They are listed in the noun resp. adjective lexicon as stems with two distinctive properties: they are non-autonomous and do not belong to a lexical category. (3) is a non concatenative rule that is used to derive abstract nouns ending with *-ist* from their corresponding adjective, for example monetär (monetary) $\mapsto$ Monetarist with the following morphological analysis: Monetarist $\mapsto$ monetär/$A \sim ist[NNSemClass = Human]$.

Rule (4) demonstrates the use of semantic typing for word formation rules. Here, the diminutive rule induced by the suffix *-chen* only applies to nouns with the semantic type ARTIFACT. This rule does not raise the recognition rate but has an impact on the precision of word formation. An example for this rule is the diminutive noun Kärtchen (engl. small card) which is derived from Karte/N $\sim$ *chen*[NN SemClass=artifact].

4 Compound Segmentation

Since morphology programs display all possible segmentations for compounds, disambiguation rules are required for compounds with ambiguous segmentations. The lexicon of complex stems is used to avoid blunders such as the segmentation of Gendarm (gendarme) into Gen and Darm or of Ration (ration) into rat$\sim$*en* and Ion by listing them as lexicalized compounds. However, it is not possible to disambiguate all compounds in that way because of the above mentioned productivity of German. Hence other methods are required to choose the correct lemma in the case of ambiguous compounds. Following the approach of Volk ([15]) we set weights for each segmentation boundary: segmentation costs for a linking element (/) in compounds are 2, for a derivation boundary ($\sim$) 2.5, for weak composition boundaries such as confix boundaries 5, for strong composition boundaries (#) 10. Furthermore, a change of lexical categories corresponds to 5 for a change from adjective to verb, 10 for a change from verb to noun, and 20 for a change proper noun to noun, since we consider compounds with proper nouns much less likely. The weights are additively combined along an accepting path of an automaton (cf. section 2.3) thus defining a cost function. The preferred analysis is the analysis with the least costs.

The following examples using some well known ambiguous German compounds illustrate the efficiency of this simple cost function. In example 1, the correct lemma Abteilung (department) is the one with the least value of the cost function since Abteilung is part of the lexicon of complex stems. On the other hand, the other two possibilities require decomposition and therefore get higher weights. In the second example, the compound Arbeitstag (work day) is correctly decomposed. Note here, that, similarly to other approaches, the wrong segmentation `Arbeit/N\#stag` is blocked by the TAGH-morphology since the rare noun

Stag (stay) is encoded in the lexicon as not being active in compounding. In example 3 the compound Schadstoffanreicherung (accumulation of toxic substance) the correct analysis is preferred because of unlikely category changes (from verb to noun resp. from proper noun to noun) in the other two segmentations.

Example (1):

```
Abteilung[NN SemClass=Human_Group Gender=fem Number=pl Case=*] <0>
ab|teil/V~ung[NN SemClass=Abstract Gender=fem Number=pl Case=*]<5>
Abtei/N#Lunge[NN SemClass=PhysObject Gender=fem Number=pl Case=*]<10>
```

Example (2):

```
Arbeit/N\s#Tag[NN SemClass=abstr Gender=masc Number=sg Case=nom_acc_dat] <12>
```

Example (3):

```
Schadstoff/N#an|reicher/V~ung[NN Sem=Abstract Gen=fem Num=sg Case=*]<15>
schad/V#Stoff/N#an|reicher/V~ung[NN SemClass=Abstract Gender=fem Number=sg Case=*] <25>
Schad/NE#Stoff/N#an|reicher/V~ung[NN SemClass=Abstract Gender=fem Number=sg Case=*] <45>
```

5 Technicalities

The morphology described here has been developed for 5 years. The development is based on the Potsdam FST library which is modelled after the seminal AT&T FST library (Mohri [11]) and implemented in C++. The library implements all operations of the algebra of weighted finite state transducers based on abstract semirings. The library also contains compilers for lexicons, regular expressions, replacement rules etc. The morphology transducer currently has 3.96 million states and 6.75 million transitions. It analyzes, depending on the text genre, up to 50,000 words per second. TAGH-morphology is currently used as an annotation tool for the search engine of the newspaper *Die ZEIT* (`http://www.zeit.de`) as well as for the morphological analysis of the DWDS-Kerncorpus of the project DWDS at the Berlin-Brandenburg Academy of Sciences (`http://www.dwds.de`). The DWDS-Kerncorpus is a 100 million word corpus of the 20^{th} century, balanced with respect to text genre. The recognition rate for the archive of *Die ZEIT* (40 m tokens) is 99.1%, the recognition rate for the DWDS-Kerncorpus is 98.2%. An evaluation of the correctness has not been carried out yet due to a lack of training corpora containing manually corrected word segmentation information.

6 Conclusion and Further Work

In this work we have presented TAGH-Morphology, a system, which unlike the two-level approach does not assume to be closed under intersection, i.e. it does not require the input and output tape to be of the same length. We have shown that the system scales up to a full coverage morphology of German and that the implemented mechanism, which is based on weighted transducers, rules out most of the undesired segmentations with a best match strategy. We have also presented a way to integrate a shallow semantic types to the morphological

analysis thus allowing to compute a semantic type for compounds that are not in the stem-lexicon.

Future work will concentrate evaluation of the correctness of the system, which amounts to the creation of a manually disambiguated training corpora, as well as to use machine learning methods in order to learn the weights of the cost function.

References

1. Augst, Gerhard: Lexikon zur Wortbildung. Morpheminventar Bd. 1-3. Tübingen, 1975.
2. Cormen, T.H., Leiserson, C.L., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, 2001.
3. Courtois, B. "Dictionnaires électroniques DELAF anglais et français", in: Leclère C., Laporte E., Piot M., Silberztein M. (eds.). Syntax, Lexis and Lexicon-Grammar. Papers in honour of Maurice Gross, Lingvisticae Investigationes Supplementa 24, Amsterdam-Philadelphia : Benjamins, 2004, p. 113 – 125.
4. Geyken, A. and N. Schrader: LexikoNet - a lexical database based on type and role hierarchies. Technical Report BBAW/DWDS, Berlin, 2005.
5. Golan, Jonathan S.: Semirings and Their Applications. Kluwer, Dordrecht, 1999.
6. Haapalainen, M. and A. Majorin: Gertwol: Ein System zur automatischen Wortformerkennung deutscher Wörter. Lingsoft, Inc., 1994.
7. Hopcroft, J.E. & Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading, Mass., 1979.
8. Kaplan, R.M., Kay, M.: "Regular Models of Phonological Rule Systems". Computational Linguistics, 20(3), 1994, p. 331–378.
9. Karttunen, L.: "Constructing Lexical Transducers". In: Proceedings of the Fifteenth International Conference on Computational Linguistics. Coling I-94, Kyoto, Japan, 1994, p. 406–411.
10. Klappenbach, Ruth and Wolfgang Steinitz (ed.) (1977). Wörterbuch der deutschen Gegenwartssprache (WDG). Akademie Verlag.
11. Mohri, M.: "Semiring Frameworks and Algorithms for Shortest-Distance Problems". Journal of Automata, Language, and Combinatorics 7 (2002) 3, p. 321–350.
12. Pustejovsky, J., P. Hanks, and A. Rumshisky. "Automated Induction of Sense in Context". 5th International Workshop on Linguistically Interpreted Corpora (LINC-04), Coling, 2004.
13. Riley, M.: "The Design Principles of a Weighted Finite-State Transducer Library". Theoretical Computer Science, 231 (2000), p. 17–32.
14. Sproat, R.: Finite-State Methods in Morphology, Text Analysis and the Analysis of Writing Systems. ROCLING X, 1997.
15. Volk, M.: "Choosing the right lemma when analysing German nouns". In: Multilinguale Corpora: Codierung, Strukturierung, Analyse. Jahrestagung der GLDV 11, Frankfurt, 1999, p. 304–310.

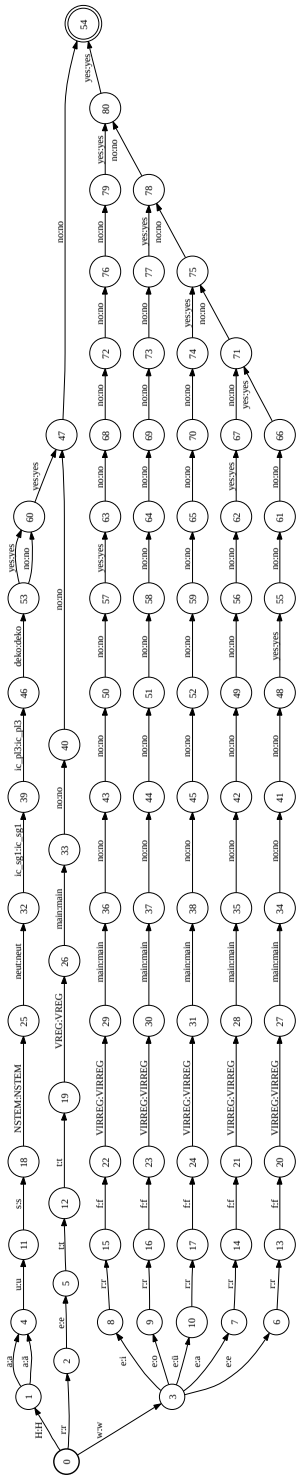

Fig 3. Stem lexicon as a transducer

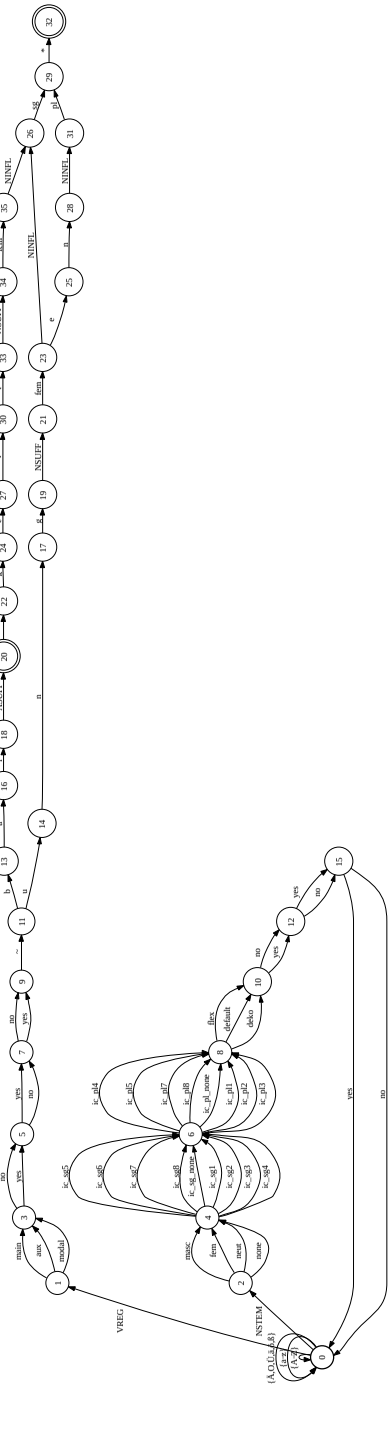

Fig.4. FSA for the grammar fragment of Fig. 2