An optimised FS pronunciation resource generator for highly inflecting languages

Dafydd Gibbon[*], Ana Paula Quirino Simões[+], Martin Matthiesen[#]

[*]Universität Bielefeld
Fakultät f. Ling. & Lit.
Postfach 100 131, D–33501 Bielefeld, Germany
gibbon@spectrum.uni-bielefeld.de

[+]CSLI, Stanford University
CA 94305-4115, USA
aquirino@stanford.edu

[#]Lingsoft, Inc.
Tehtaankatu 27-29 D
FIN-00150 Helsinki
Finland

Abstract

We report on a new approach to grapheme–phoneme transduction for large-scale German spoken language corpus resources using explicit morphotactic and graphotactic models. Finite state optimisation techniques are introduced to reduce lexicon development and production time, with a speed increase factor of 10. The motivation for this tool is the problem of creating large pronunciation lexica for highly inflecting languages using morphological out of vocabulary (MOOV) word modelling, a subset of the general OOV problem of non-attested word forms. A given spoken language system which uses fully inflected word forms performs much worse with highly inflecting languages (e.g. French, German, Russian) for a given stem lexicon size than with less highly inflecting languages (e.g. English) because of the 'morphological handicap' (ratio of stems to inflected word forms), which for German is about 1:5. However, the problem is worse for current speech recogniser development techniques, because a specific corpus never contains all the inflected forms of a given stem. Non-attested MOOV forms must therefore be 'projected' using a morphotactic grammar, plus table lookup for irregular forms. Enhancement with statistical methods is possible for regular forms, but does not help much with large, heterogeneous technical vocabularies, where extensive manual lexicon construction is still used. The problem is magnified by the need for defining pronunciation variants for inflected word forms; we also propose an efficient solution to this problem.

1. Morphologically based grapheme–phoneme transduction

In the work reported on here,[1] a new approach to grapheme–phoneme transduction for large-scale German spoken language corpus resources using explicit morphotactic and graphotactic models is presented. Finite state optimisation techniques are introduced to reduce lexicon development and production time, with a speed increase factor of 10.

The motivation for this tool is the problem of creating large pronunciation lexica for highly inflecting languages using morphological out of vocabulary (MOOV) word modelling, a subset of the general OOV problem of non-attested word forms. A given spoken language system which uses fully inflected word forms performs much worse with highly inflecting languages (e.g. French, German, Russian) for a given stem lexicon size than with less highly inflecting languages (e.g. English) because of the 'morphological handicap' (ratio of stems to inflected word forms), which for German is about 1:5.

However, the problem is worse for current speech recogniser development techniques, because a specific corpus never contains all the inflected forms of a given stem. Non-attested MOOV forms must therefore be 'projected' using a morphotactic grammar, plus table lookup for irregular forms. Enhancement with statistical methods is possible for regular forms, but does not help much with large, heterogeneous technical vocabularies, where extensive manual lexicon construction is still used.

The problem is magnified by the need for defining pronunciation variants for inflected word forms; we also propose an efficient solution to this problem.

The present approach to grapheme–phoneme transduction was developed for a morphologically structured lexicon of German for use in speech–to–speech translation. In the lexicon construction process, large scale corpus analysis of transcriptions of spoken appointment scheduling dialogues was required in the context of an acquisition system for morphological parsing and classification for the insertion of lexical stems into an inheritance lexicon. For large–scale analysis an important criterion is operational efficiency. One of the central functions of the lexicon is to feed a morphological paradigm generator for producing fully inflected word forms for use in speech recognition and synthesis. The task is therefore to acquire attested words from the corpus together with their morphological structure and classification and with a phonological (actually morphophonological) representation suitable for driving the paradigm generator.

The architecture of the lexicon acquisition system is shown in Figure 1.

2. The morphophonological transduction problem

The suitability of various statistical and symbolic rule–based approaches to grapheme–phoneme conversion were considered (Klenk and Langer, 1989), and it was concluded that because of the strong morphological conditioning of the grapheme–phoneme relation in German it was necessary to include morphological criteria in the converter specification:

1. German spelling is closer to morphophonological than phonemic representation (lack of correlates for final devoicing, de-rhotacisation etc.).

[1]The MCLASS classifier component was developed by Harald Lüngen, Universität Bielefeld.

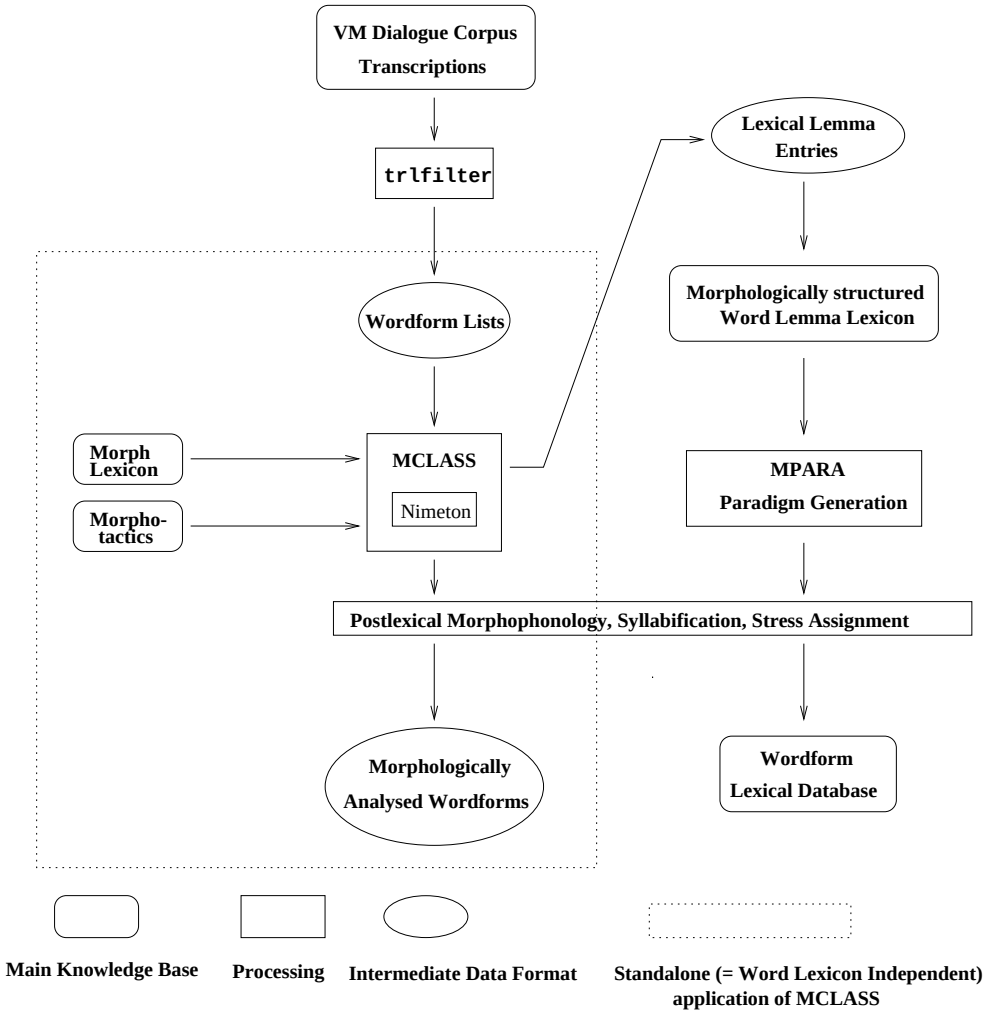

Figure 1: Morphological corpus processing environment.

2. Stem vowels and, in some cases consonants, alternate according to morphological criteria.

3. Contexts for phonological alternations are defined by inflectional contexts (e.g. realisation of /r/, final devoicing of /b,d,g,v,z/ to /p,t,k,f,s/, spirantisation of /g/ to /C/).[2]

Consequently it was necessary to use a *morphophonological* underlying representation for the phonology, not a *phonemic* representation. For example, *Sieb* sieve" is not represented as /zi:p/, with final devoicing, but as /zi:b/ in order to have a general stem form for use in generating all elements of the inflectional paradigm (in the generator, morphophonological rules are cascaded after the morphotactic generation component); cf. (Bleiching et al., 1996). Since this kind of information is not directly flagged in the orthography, a morphological parser with the capability of segmenting and classifying affixes and stems for insertion into the paradigm generator lexicon was developed. The overall parser architecture (MCLASS) is shown in Figure 2; cf. (Lüngen et al., 1998), (Steinbrecher, 1995) and (Lüngen and Sporleder, 1999).

A further novel feature of this parser is that morphological roots (lexical morphemes, bases) are also explicitly parsed using a graphotactic model, in order to capture the full range of 'out–of–vocabulary' items: constraints on the well–formedness of roots are needed in order to limit the search space. The grapheme–phoneme relation is very heavily conditioned by morphophonological structure, i.e. phonotactics, and a graphotactic network was developed by analogy with phonotactic networks which have been developed for phonology (cf. (Carson-Berndsen, 1998)). This made it possible to combine morphological, graphotactic and phonotactic constraints in one homogeneous transducer in defining the grapheme–phoneme relation.

It was hypothesised that a suitable way of expressing the grapheme–phoneme relation for morphological roots, and incorporating both morphological and phonological constraints, was to use a finite–state transducer, since the use of finite state techniques in morphology is well established (Koskenniemi, 1983; Karttunen, 1983) as is their use in phonology (Kaplan and Kay, 1994; Carson-Berndsen, 1998). Several approaches to solving the problem of in-

[2]Where necessary the SAMPA conventions for IPA symbols are used, cf. (Gibbon et al., 1997).

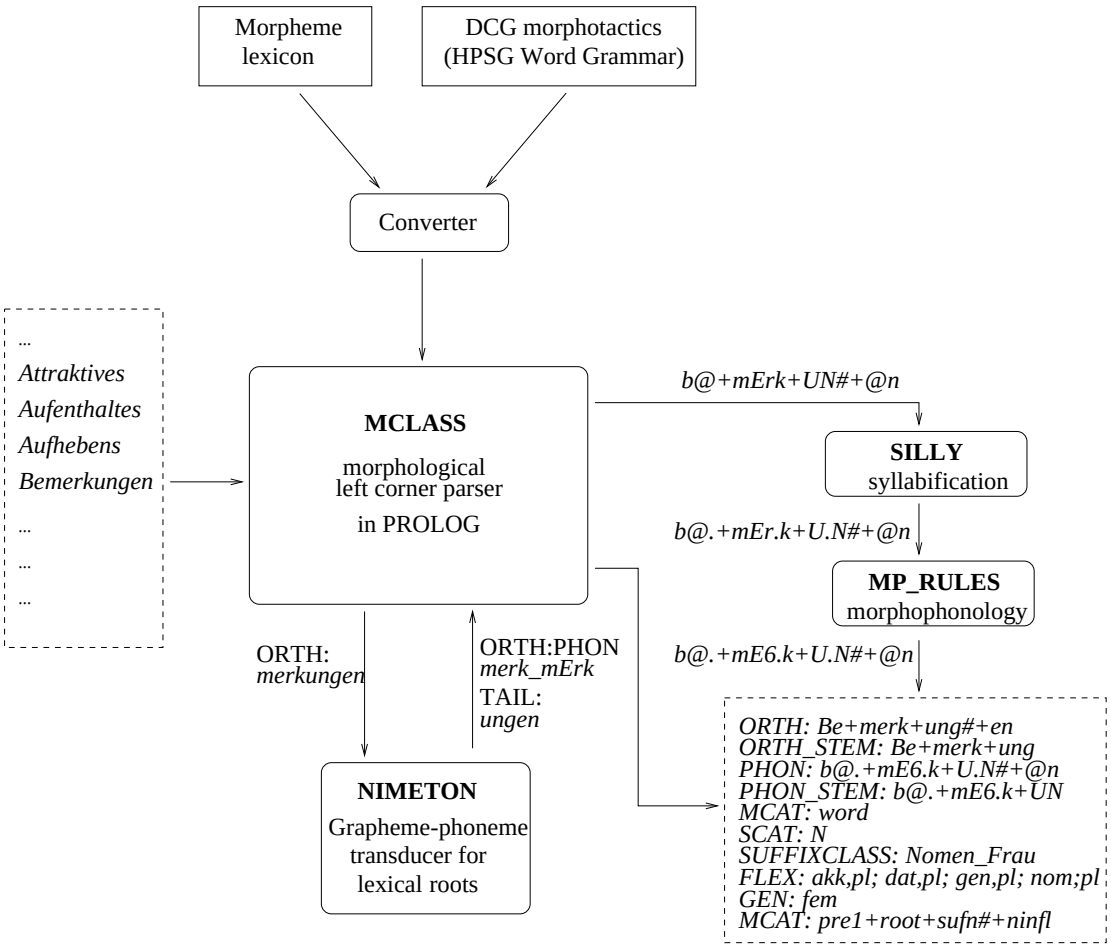

Figure 2: Morphological parser architecture.

terfacing phonology and morphology were considered, including Two–Level Morphology and Lexical Phonology (Kaisse and Hargus, 1993), and a finite–state model based on principles of Lexical Phonology was developed. The relevant principles of Lexical Phonology are:

1. More abstract affixation levels in Lexical Phonology define affixes which are closer to the stem.
2. Phonological rules are specific to morphological levels.

On closer inspection, first, it quickly became apparent that the 'closer–to–the–stem' criterion simply meant a specific linear order, and is easily modelled in straightforward finite state terms:

$(\text{AFFIX}_n)^*$ ... $(\text{AFFIX}_1)^*$ ROOT $(\text{AFFIX}_1)^*$ ... $(\text{AFFIX}_n)^*$

where the subscripts denote the Level number in Lexical Phonology. On this analysis it is no accident that root operations also operate at level 1 in interdependence with the Level 1 affixes.

Second, the pairing of morphological and phonological rule sets at well–defined levels was re–formulated as a principle of compositionality in phonology: the phonology of a complex word is a (morphophonological) function of the phonology of its parts.

The following operations are performed by the parser:

1. Morphological root transduction.
2. Derivational prefix and suffix transduction with boundary assignment.
3. Inflectional affix transduction with boundary assignment, syllabification and morphophonological rule cascade.
4. Handling of compounds with concatenation of stems or inflected stems.

One novel part of the parser, the morphological root transducer and its implementations NIMETON and X-NIMETON, is described below.

3. Design considerations

The tool has a novel architecture which is described declaratively as a tuple $< SL, MS, PH, AS, WS, PM, MI >$: stem lexicon; morphosyntactic category set; paradigm hierarchy; affix set; word set (fully inflected forms); paradigm mapping $PM : SL, MS, PH- > AS$; interpretation function $MI : SL, AS- > WS$. AS is the domain of PM; MI is thus a composite function taking stem lexicon and

paradigm map values AS as arguments (Bleiching et al., 1996). MS categories follow accepted tagset standards, and include inflectional and derivational categories; AS contains inflectional and derivational affixes; WS contains inflected and derived words.

Pronunciation variants are generated using a phonotactic rule set PR and a phonetic interpretation function PI:WS,PR-¿VS from wordforms into a variant set.

The grapheme–phoneme relation in German is heavily conditioned by morphophonological structure, unlike languages with straightforward letter-to-sound functions. The theoretical linguistic model with the closest specifications, Lexical Phonology, was found to be unnecessarily complex: we were able to reduce the basic principle of more 'abstract' affixes being 'closer' to the stem to a regular expression

$$Prefix_n * ... Prefix_1 * Stem ... Affix_1 * ... Suffix_n *$$

.

This immediately maps into an FST, there being are only two interesting dependencies between prefixes and suffixes (*ge ... pastpart*, and *zu...infinitive*), both of which are FS-tractable.

The functions PM, MI, PI are operationalised with finite state transducers (FST); function composition is operationalised using the cascaded FSTs (Kaplan and Kay, 1994). The protype was implemented in Sicstus Prolog with a straightforward nondeterministic tail-recursive algorithm.

The data structures for the FSTs are phonotactic networks formulated as a four place relation (following Carson-Berndsen 1998). Three lexical root networks lexical roots: for native roots and proper names, and a stress-assignment network. The root networks are concatenated in the FST described above.

The morphological root transducer By generalising over prefixes and suffixes of a list of roots extracted from various corpora and lexica, a graphotactic finite state transducer was developed in order to specify the notion of *possible orthographic root of German*. The transducer for German native root morphs is represented in transition network notation in Figure 3.

The essential difference between the present approach and a two–level type approach to specifying the transducer is that the transducer constitutes a single integrated *morphological root grammar*, and not a set of separate filters or constraints over possible neighbours in restricted linear windows. The transitions are labelled by grapheme–phoneme pairs; the topology of the network clearly shows the distribution of constraints over the whole root, and consequently captures the context–specific conversion relations of graphemes in different positions in the root. Because of alternative pronunciations of a grapheme at the same position in the root, the transducer is non–deterministic.

The notation used to mark morphological boundaries in the output of the transducer is shown in Table 1.

A number of heuristic decisions were taken:

1. Vowels marked as graphemically long, such as <ie, ah, oo> are always transduced as long; if a contradiction between this graphemic constraint and morphophonological constraints occurs, the graphotactic constraint *overrides* the phonotactic constraint.

2. Vowels before double consonants are transduced as short vowels, for example <kann>; lengthening is not permitted at this point.

3. Vowels before single–grapheme consonants are transduced as long vowels, for example <ros> in <Ros#+e>, <a> in <abend>.

4. The grapheme <e> is transduced to /@/ if it is the second vowel grapheme of the root, e.g. in <roden>.

Table 1: Morphophonological boundary marker conventions.

Marker	Description
#	Word boundary (e.g. between constituents of compound words)
+	Derivational affix boundary (e.g. between affixes and affixes, or affixes and stems)
#+	Inflectional affix boundary between stem and affix
.	Syllable boundary

Pronunciation variant FST. Rules for pronunciation variants (Kirchhoff, 1995) were formulated as an FST cascade following (Kaplan and Kay, 1994), which was cascaded with the morphophonological FST. The composition procedure defined by Kaplan and Kay can in principle be used to create a single overall automaton from the morphophonological and pronunciation variant cascade; in practice, automaton size limitations may prohibit this.

4. Implementation: NIMETON

The innovative feature of the present implementation, as against two-level FS morphology, is that the networks represent a complete, homogeneous grammar, and not a set of essentially unrelated coordinated constraints.

The 'naive' prototype implementation was used for about a year for pronunciation lexicon generation in a speech-to-speech translation project, with fully inflected form dictionaries of approximately 50k words, derived from a stem dictionary of about 10k stems. The dictionaries were evaluated with an independently developed phonotactic recogniser and by trained phoneticians.

Slow processing and increasing vocabulary size made a fresh start necessary. We took a novel approach for this domain by porting the FSTs to the Xerox `xfst` toolkit. The SAMPA-based FST notation was converted to `xfst` import notation, exception automata were created from the lookup tables, and overgeneralising exception paths through the main automaton were removed using the difference operation in the xfst calculus. The exception automata as a whole were unified with the main automaton. The FSTs were determinised, taking input-output pairs as single symbols, resulting in a deterministic, pruned, minimised, epsilon-free and loop-free automaton. Pronunciation variant automata

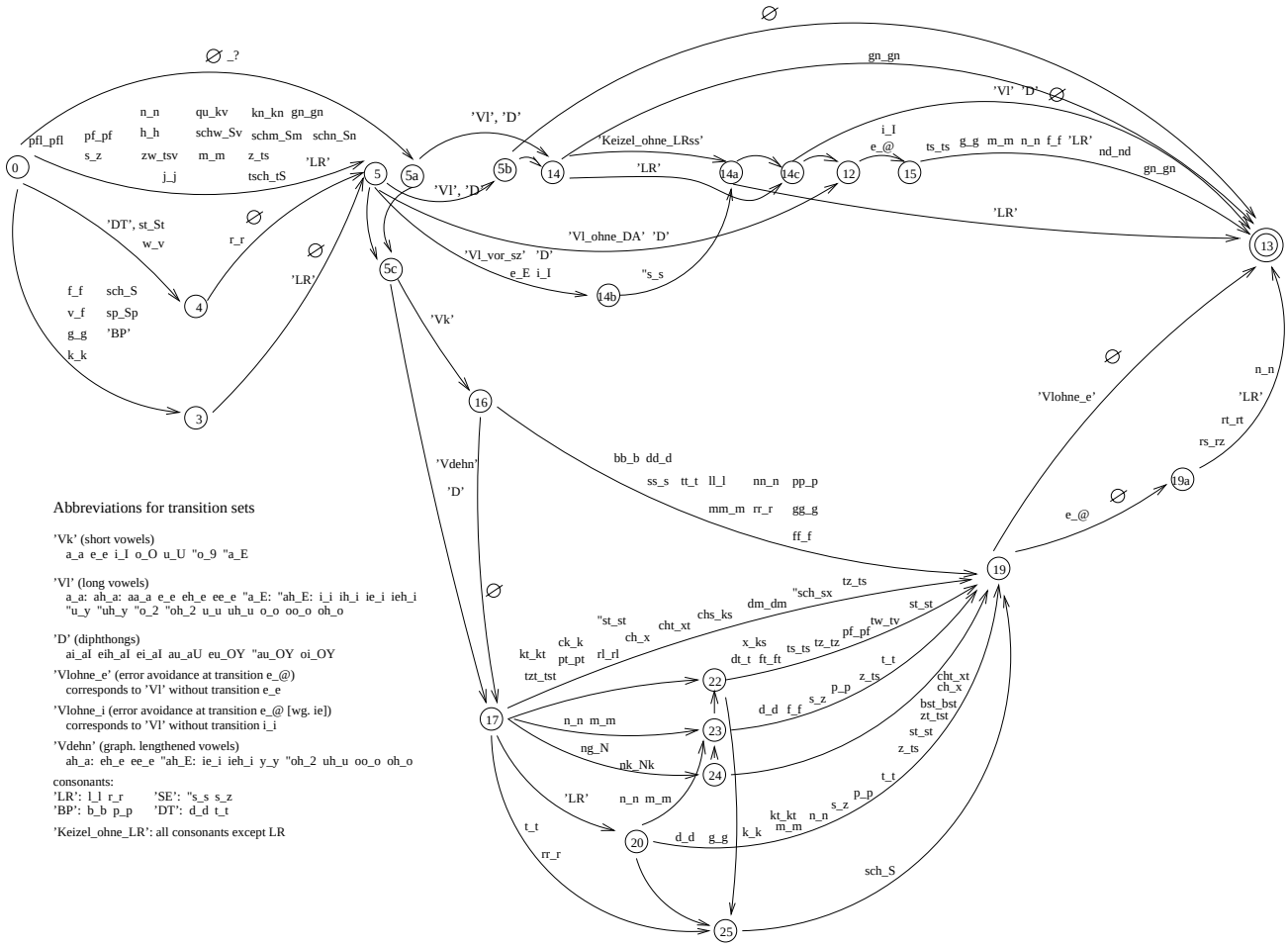

Figure 3: Graphotactic transition network for root.

were developed on the same lines, but not composed with the main automaton because of size limitations.

The initial implementation of the root transducer, (NIMETON (Matthiesen, 1996)) was designed in Prolog. The implementation consists of three distinct components. The main network encodes the syllabic structure of German roots (orthographic–morphophonological morphs), and morphs of foreign origin and non–native orthography are currently handled as exceptions and listed with their phonological representations; this inventory is consulted before finite state processing. A schema of the internal structure of Nimeton is shown in Figure 4.

Nimeton is constituted by 3 finite state nets, that can be accessed separetedly, each one having a different function: The main net, called `net--nat`, is responsible for the transduction of the German native lexical root. The second one, `net--pn`, with a slightly different structure, is able to transduce German proper names. The third automaton, `net--str`, actually very small, adds to each transduced phonemic representation a standard stress marking. The interaction between the three nets can be observed in the figure above.

The use of Prolog, the non–determinism of the network, and the introduction of non–determinism for reasons of compactness and linguistic plausibility led to rather slow performance, resulting in low overall response speed when embedded in the overall morphological transduction system.

Table 2: Nimeton evaluation results.

Transduction equivalent to source:	1764	93.9%
False transduction:	81	4.3%
No transduction:	33	1.8%

When the graphemic-phonemic transducer was developed, accuracy rather than efficiency was initially the main concern. The basis for the evaluation was a list with 1878 lexical roots taken from a data base of full forms (Ehrlich and Gibbon, 1995); different error types were flagged.

The evaluation results are shown in Table 2.

5. X–NIMETON

In order to increase the efficiency of the system, two steps were required:

1. Maximal determinisation of the transducer.
2. Implementation in a more efficient language.

It was decided to use the Xerox `xfst` system to re–implement NIMETON, for both these reasons: `xfst` in-

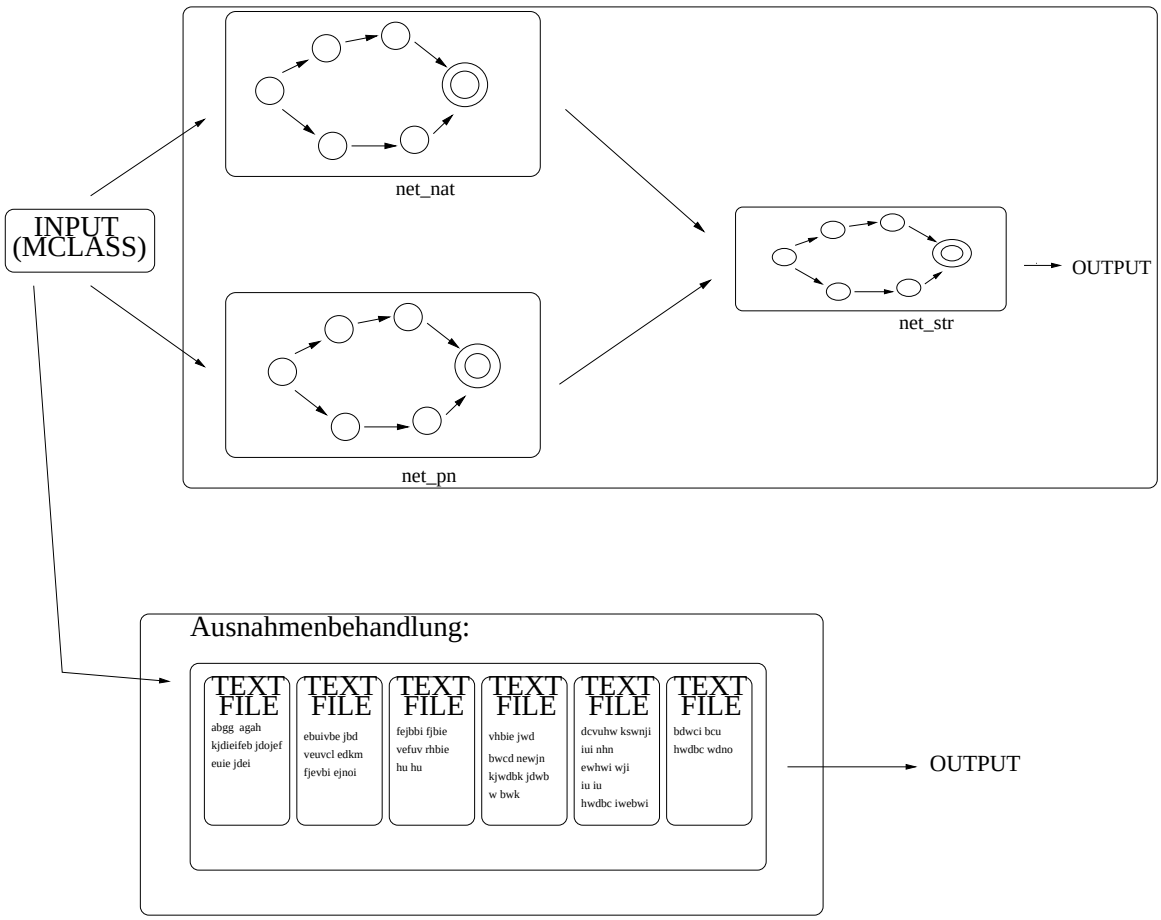

Figure 4: Internal structure of NIMETON.

corporates a determinisation algorithm and an efficient finite state transducer compiler in C.

The internal structure of the Prolog implementation of Nimeton was already discussed in the section above. In order to work with it using the Xerox `xfst` system it was necessary to restructure the interaction of the independent nets, which were formulated as Prolog clauses in the original implementation. Therefore the different finite state nets were first exported into the Xerox `xfst` system using the `xfst` Prolog interface. Then, the independent nets were combined in the following way:

1. `net--nat` was composed with `net--str`;
2. `net--pn` was composed with `net--str`;
3. the two automata that resulted from these operations were unified.

After these steps the main part of X-Nimeton, which encodes the generalisations of the German syllabic structure, was complete. The second step was to take care of the exception handling mechanisms, which were processed by an additional module in the Prolog implementation. For this purpose a new finite state automaton was generated, which recognises the morphemes of Nimetons exception lists and provides their respective phonemic representation.

As already mentioned, the Prolog implementation of Nimeton has an extra module for exception handling, which searches for each morpheme to be transduced in the exception lists making sure that the right transduction will be actually provided. In the `xfst` implementation it was not possible to create such a module, as the software can only process finite state nets. In order to make sure that the main part of X-Nimeton cannot provide any false transduction of morphemes which have to be taken from the exception lists, all paths which would recognise graphemes listed as exceptions were removed from the main part of X–Nimeton by the difference operation. This step guarantees that no exception will be handled as a native root and has the same effect as the exception handling module of the Prolog implementation.

The next step was to unify the main net (without the exception paths) and the exception net.

Finally, the resulting structure was determinised as an automaton, taking the input–output tape pairs as a single symbol for this purpose. The number of states was minimised, and all loops and epsilon transitions as well as all paths leading to non–final states were removed, resulting in a deterministic, pruned, minimised, epsilon–free and loop–free automaton, which is equivalent to the Prolog implementation.

6. Evaluation and conclusion

We have described a new approach to grapheme–phoneme transduction for large scale corpus analysis in lexicon acquisition for a speech–to–speech translation system. Unlike conventional grapheme–phoneme transducers, which are generally either statistically trained or based on direct grapheme–phoneme translation rules, or on a mixture of these, our approach uses an explicit and empirically complete model of German morphotactics derived from Lexical Phonology, and, a further novel feature, an explicit graphotactic model of German morphological roots.

The prototype was implemented as a finite state transducer interpreter in Sicstus Prolog. In view of the large amounts of data to be processed during evaluation of the transducer, it was decided to use the optimisation techniques of current finite state technology, and the system was ported to the Xerox `xfst` system.

The automaton was successfuly tested for exact (100%) I/O equivalence with the previously validated Prolog automaton. The final automaton consists of 74 states, 1062 arcs, and 3.893171e6 (nearly 4 million) paths. The system was benchmarked 10 times faster than the compiled Sicstus Prolog prototype.

The pronunciation variant cascade will be discussed in a separate paper.

We suggest that our method can be profitably used as part of a strategy for overcoming the current pronunciation lexicon bottleneck problem for speech and multimodal systems alongside statistical methods and human pronunciation lexicon construction.

7. References

Bleiching, Doris, Dafydd Gibbon, and Guido Drexel, 1996. Ein synkretismusmodell für die deutsche morphologie. In Dafydd Gibbon (ed.), *Natural Language Processing and Speech Technology: Results of the 3rd KONVENS Conference, Bielefeld October 1996*.

Carson-Berndsen, Julie, 1998. *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition. Text, Speech and Language Technology Series, Volume 5*. Kluwer Academic Publishers.

Ehrlich, Ute and Dafydd Gibbon, 1995. Spezifikation für ein Verbmobil Lexikondatenbankkonzept. Verbmobil Memo Nr. 69.

Gibbon, Dafydd, Roger Moore, and Richard Winski, 1997. *Handbook of Standards and Resources for Spoken Language Systems*. Berlin: Mouton de Gruyter.

Kaisse, Ellen and Sharon Hargus, 1993. Introduction. *Studies in Lexical Phonology (Phonetics & Phonology 4)*:1–19.

Kaplan, Ronald M. and Martin Kay, 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Karttunen, Lauri, 1983. KIMMO: a general morphological processor. *Texas Linguistic Forum*, 22:163–186.

Kirchhoff, Katrin, 1995. Two-level modelling of speech variant rules. Verbmobil Report 82. Unversitt Bielefeld.

Klenk, Ursula and Hagen Langer, 1989. Morphological segmentation without a lexicon. *Literary and Linguistic Computing*, 4, No. 4:247–253.

Koskenniemi, Kimmo, 1983. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. University of Helsinki, Department of General Linguistics.

Lüngen, Harald, Karsten Ehlebracht, Dafydd Gibbon, and Ana Paula Quirino Sim oes, 1998. Bielefelder Lexikon und Morphologie in Verbmobil Phase II. Verbmobil Report 233. Universtität Bielefeld.

Lüngen, Harald and Caroline Sporleder, 1999. Automatic induction of lexical inheritance hierarchies. In Jost Gippert (ed.), *Multilinguale Corpora. Codierung, Strukturierung, Analyse*. Prague: Enigma Corporation.

Matthiesen, Martin, 1996. Nimeton - Ein Graphem-Phonem-übersetzer für das Deutsche. VERBMOBIl Memo 109.

Steinbrecher, Daniela, 1995. MSEG: Morphologische Segmentierung deutscher Wortformen. Verbmobil Memo 99. Unversität Bielefeld.