BMBF

Verbmobil
Verbundvorhaben

Two-level Modelling of Speech Variant Rules

Katrin Kirchhoff

Universität Bielefeld

Vm
Report 82
Mai 1995

Mai 1995

Katrin Kirchhoff

Universität Bielefeld (UBI)
Fakultät für Linguistik und Literaturwissenschaft
Universitätsstr. 25
Postfach 10 01 31
33501 Bielefeld

Tel.: (0521) 106 - 3510
Fax: (0521) 106 - 6008
e-mail: `katrin@Spectrum.Uni-Bielefeld.DE`

Gehört zum Antragsabschnitt: 15.6 Interaktive Phonologische Interpretation

Die vorliegende Arbeit wurde im Rahmen des Verbundvorhabens Verbmobil vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) unter dem Förderkennzeichen 01 IV 101 B 2 gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei dem Autor.

Abstract

This paper describes a phonetic knowledge base for German consisting of a set of speech variant rules. These rules have been established on the basis of empirical, corpus-based investigations enriched by linguistic generalisations. Theoretical and computational foundations of speech variant rules are discussed, and their practical application in a linguistic word recognition system (*BELLEx3*, U Bielefeld) is demonstrated. Although the speech variant rules described in this paper have been established for the purpose of knowledge-based word recognition, their declarative implementation in a two-level transducer enables them to be employed for both recognition and generation of speech variants. Finally, an extension of standard two-level techniques is described whereby two-level transducers defining constraints on mapping relations between input and output forms are integrated with wellformedness-constraints on input forms stated in terms of finite-state automata.

Contents

1 Introduction

One of the most difficult problems in automatic speech recognition (ASR) is the high degree of phonetic variability which continuous speech typically exhibits. Words are assumed to have a so-called standard or *canonical* pronunciation, i.e. that form which is listed in pronouncing dictionaries, usually corresponding to isolated, clear speech. In continuous speech, however, these forms are rarely to be found due to assimilation and coarticulation phenomena. This complicates the task of mapping the speech signal onto a sequence of lexical representations.

Currently, there are two major approaches to speech recognition, which can be characterised as *statistical* versus *rule-based*. Statistical systems aim at recognising speech on the basis of algorithms of pattern classification and stochastics, the most widely used technique being hidden Markov modelling. The rule-based approach of Artificial Intelligence and computational linguistics tries to make explicit the linguistic knowledge involved in human speech recognition in order to use implementations of this knowledge in ASR systems.

The level of phonetic realisation and the level of lexical representation (which, in this case, is assumed to be the canonical phonemic representation) are related not by random variation but by highly regular phonetic knowledge. A description of this knowledge in terms of phonetic rules can be beneficial to both the statistical and the rule-based approach.

In statistical systems speech is recognised by calculating the maximum probability of a sequence of acoustic observations corresponding to a given model. Most current statistical ASR systems use phoneme models as their basic units. However, some of them, most notably automatic labelling systems, operate directly on word models. If the pronunciation lexicon used for this task merely includes canonical pronunciations, or canonical forms enriched by the variants found in the training corpus, the system will not correctly recognise unattested (new) variants. Thus, rules can be employed to generate all possible variants of canonical forms in order to expand the lexicon, which then includes not only the canonical forms (and possibly the variants occurring in the training corpus), but also the remaining set of possible but unattested non-canonical forms. This expansion of the lexicon should in turn yield better labelling results.

Rule-based systems may use phonetic rules directly during the recognition process, mapping the output from a front-end phoneme recogniser to a lexicon. This has been referred to as the *analytic* use of rules whereas the former method can be described as the *generative* use (Hoequist & Nolan (1991)).

This paper describes a set of speech variant rules for German, its empirical, linguistic and computational foundations, and potential as well as existing applications, in particular its integration into a linguistically-based word recogniser.

Two major requirements are imposed on speech variant rules: First, although the rules are to be employed for the specific purpose of recognition, they should nevertheless form a *declarative* knowledge base. This means that the form of the representation of phonetic knowledge should be independent of the way in which this knowledge is used. A declarative representation has the advantage of being compatible with both generative and analytic applications; it is therefore reusable. Second, such a knowledge base should be largely speaker-independent and corpus-independent in order to attain a nearly complete coverage of the regular speech variants of Standard German.

The rest of this paper is organised as follows: In section 2 the data and method used to establish the speech variant knowledge base are presented. Section 3 will focus on linguistic aspects of speech variant rules, such as the various rule types, the representation of segments and rules, etc. Section 4 will consider computational issues, such as the formal device needed to implement speech variant rules and the question of declarativeness. Potential and existing applications are described in section 5. The rules themselves will be given in section 6.

2 Data and Method

There are several types of variation which have to be distinguished in the context of speech recognition:

- **acoustic variation**
  The term acoustic variation refers to the different physical manifestations of sound tokens belonging to the same *phonetic* category. Acoustic variation is found both across speakers and within speakers and depends on such factors as the recording conditions, the vocal tract size of the speaker, etc.

- **allophonic variation**
  In its narrow sense, the term allophonic variation refers to those effects on segments which are brought about by their phonetic context and which do not entail a categorical *phonemic* change in the hearer's perception of the sound. As an example, the /k/ sounds in /kYC@/ and /kats@/ will differ considerably in their phonetic realisation, the first having a more palatal and the second a more velar quality, as conditoned by the quality of the preceding vowel. However, both are assigned to the phonological category /k/.

- **phonotypical variation**
  The change effected on segments by coarticulation and assimilation as described above is always gradual. However, a point may be reached at which the coarticulated sound is perceived as belonging to a different phonemic category. Consider the assimilation of the alveolar nasal: when preceding a labial consonant the alveolar nasal /n/ may be shifted in its place of articulation to approximate that of the following labial. The result may be a labiodental nasal. However, since German does not possess phonemic labiodentals the sound in question will be perceived as a bilabial nasal, which does form part of the phonological system of German. Thus, /zEnf/ (*Senf*) may become [zEmf]. In short, phonotypical variation is that type of variation which can be expressed by means of the phonemic sound inventory of the language.

Here, we will exclusively concentrate on phonotypical variation, i.e. those sound changes which affect the phonemic identity of sounds and which can therefore be captured by segmental labelling of speech files. It should be noted that although most of the phonotypical variation in speech data can be captured by rules, there are variants which are better treated differently. This concerns most of all the behaviour of function words, which are often reduced to an extreme degree. However, since function words are closed-class words their variants can simply be listed in the lexicon.

The rules described below have been established on the basis of a systematic analysis of several speech corpora; they thus have an empirical foundation. The corpora used were the following:

1. EUROM.1 Database
The EUROM.1 database was recorded in 1992 at the University of Bielefeld as part of the ESPRIT project 2589 (*Multi-lingual Speech Input/Output Assessment, Methodology and Standardisation - SAM*). The recordings consist of numbers, CVC words and texts, which were read aloud by speakers from a prompting monitor. For the present investigation 48 manually labelled recordings of text passages were used, the length of which varied between 16 and 24 seconds. Texts were labelled on a broad phonetic (phonotypical) basis. The transcription alphabet was German SAMPA (Wells (1989)).

2. PhonDatII Database
The PhonDatII database, recorded in 1992 at the Universities of Kiel, Bonn and Munich, is a corpus of train connection requests consisting of 200 utterances per speaker. The utterances, which were presented to speakers in the form of screen prompts, had been developed on the basis of spontaneous dialogues. All 200 ut-

terances were read by all 16 speakers. 64 utterances per speaker were labelled on a broad phonetic basis, the remaining utterances were labelled at word level. In addition to this, the 136 word-labelled utterances of one speaker (SAT) were labelled segmentally. The alphabet used was a modified version of SAMPA.[1] For the present investigation, all segment-labelled files were taken into account.

3. VERBMOBIL Speech Data (Blaubeuren Dialogues)
The VERBMOBIL speech data consists of scheduling task dialogues between two interlocutors. Ten of these dialogues, recorded at the University of Karlsruhe, were used for the present task. The corresponding segmental label files were produced automatically and subsequently corrected manually. The total number of turns investigated was 205; the length of turns varied between 2 and 14 seconds.

All segment label files were then compared with the canonical transcriptions of the utterances and non-canonical forms were extracted.[2] These were subsequently grouped into two categories: rule-based and irregular variants. Rule-based variants are those forms which have a plausible phonetic basis, such as contextual assimilation, or which are caused by extra-phonetic but equally regular phenomena such as spelling pronunciation. All these forms are speaker-independent and recurrent. An example of a rule-based variants is the form [gu:d@] for /gu:t@/ (*gute*); voiceless consonants tend to become voiced between two sonorants. Irregular variants, on the other hand, occur only sporadically. They usually do not have a regular phonetic or extra-phonetic basis, and they are non-recurrent. The form [hambUIk] for /hambU6k/ (*Hamburg*) is an example of the category of irregular variants. The palatalisation of the diphthong does not seem to be a phonetically-based process but a spontaneous production. Only the regular variants were further considered for the generation of phonetic rules. In view of the fact that our database is to be used for the mapping from phoneme strings to lexical representations, subphonemic phenomena, e.g. the regularities underlying the nasalisation of vowels or the glottalisation of voiceless plosives, were ignored.[3]

[1] The modified SAMPA alphabet additionally includes symbols for glottalisation and nasalised vowels.

[2] Since the canonical transcriptions were established by different groups for the different corpora they are inconsistent with respect to the transcriptions of certain lexical items. Nevertheless, these inconsistencies did not affect the actual rules, which are based on sounds or sound classes instead of words.

[3] This does not mean that these phenomena should generally be ignored in speech recognition. Though irrelevant for the mapping between phonemes and lexical representations, they are often dependent on the segment's position in the syllable and can therefore be used in order to detect constituent boundaries (c.f. Church (1987)).

On the basis of these regular phonetic variants generalisations were established concerning the type and the context of speech variants. In most cases, speech variant rules do not apply to single segments but to entire classes of segments, such as the class of non-coronal nasals or the class of voiceless consonants. These generalisations were then formulated as phonetic rewrite rules.[4] Taking the above example of voicing assimilation, the corresponding rule would be:

(1) $[\text{-voi}] \rightarrow [\text{+voi}]\ /\ [\text{+voi}]\ \underline{\hspace{2em}}\ [\text{+voi}]$

The decision whether the data available allow such generalisations to be made is not always straightforward. Consider the phenomenon of initial devoicing: stem-initially, devoicing of consonants may take place. However, this applies primarily to the voiced coronal fricative /z/, although our databases also contain instances of devoicing of /b/ and /d/, for example. If the rule of initial devoicing were generalised to all voiced consonants some highly unlikely non-canonical forms would have to be postulated. In cases like theses, practical considerations were taken as the criterion on the basis of which rules were included in the knowledge base: generalisations which would produce too many unacceptable non-canonical forms (or, inversely, which would enlargen the search space for canonical forms too much) were omitted. For this reason, our knowledge base may not be able to recognise or generate *every* possible variant which might occur in Standard German but, on the other hand, a high degree of overgeneration is avoided.

As will be discussed in section 4, a set of (sequential) rewrite rules is in many ways inadequate as the basis of an implementation of pronunciation variants. Therefore, the concept of two-level transducers was chosen to implement the rules. For each rewrite rule a corresponding transducer was specified. These were then grouped into constituent networks according to their domain of application within the word. Finally, they were compiled into one single transducer. The concepts and steps involved in this process will be described in greater detail in sections 3 and 4.

3 Linguistic Aspects of Speech Variant Rules

Let us first have a look at the various types of rules and at the treatment they require. Rules can be classified according to various criteria:

[4] Similar investigations have produced phonetic rules which largely overlap with the rules presented in this paper (e.g. Kohler (1974), (1979), (1990)). Here, however, care has been taken to additionally analyse the (morphological and phonotactic) domain of application for every speech variant rule and to integrate these different types of information into a coherent system.

- context-sensitive v. context-free
- the domain of application
- the type of change effected by the rule

Context-sensitivity

Most phonetic rules operate only in a specific context. The most obvious example of this rule type is assimilation, where a sound adopts the voicing, manner or place features of one or more adjacent sounds. The concept of context can be extended to include the prosodic context, i.e. the absence or presence of a lexical accent on that syllable. Context-free rules, on the other hand, may be operative regardless of the (linear or prosodic) context in which the segment appears. An example of this type of rule is glottal stop dropping. In German, vowels at the onset of stressed syllables are preceded by a glottal stop in standard pronunciation, which can be dropped at the level of phonetic realisation. Although the distribution of the glottal stop itself is limited to a specific context, its deletion is not, as it can take place wherever a glottal stop is present in the input form. This is independent both of whether the syllable has main or secondary stress and of the preceding or following segments. Naturally, certain contexts are more likely to trigger glottal stop deletion than others: word-medially, glottal stops are more likely to be dropped than, say, at the beginning of an utterance (see Kohler (1994)) for an analysis of these tendencies). However, similar tendencies can be found for any speech variant rule. They may be exploited as additional constraints (see section 7), but they do not have rule character as they do not specify necessary conditions for the occurrence of variants.

Domain of Application

Furthermore, rules can be grouped according to their domains of application. Speech variant rules are for the most part limited to a certain constituent, either at syllable-level or at word-level. Glottal stop dropping, for instance, is limited to the syllable onset because the glottal stop only occurs syllable-initially. Plosive epenthesis is restricted to the syllable coda; schwa deletion only occurs stem-finally but not, for example, in prefixes. Velar fricativisation takes place stem-finally. The case of voicing assimilation is more complicated. As mentioned above, a voiceless consonant may become voiced between two voiced sounds. We thus have to take into account all possible environments where two voiced sounds may come together. This only happens when two syllables are joined because in the German coda all voicing distinctions are neutralised in favour of voicelessness. In the onset, clusters of at most three consonants are allowed, of which only the final consonant may be voiced. Thus, neither in the onset nor in the coda can a

sequence of [+voice] [-voice]
[+voice] possibly arise. However, several types of syllable junctions can produce a voiceless sound between two voiced sounds; therefore, all these types have to be marked as possible domains for this rule.

Further constituents which were found to be domains for the operation of speech variant rules are: stem-initial onset, syllable onset, nucleus, coda, rhyme, stem-final coda, stem-final syllable, consonant junction, stem-final consonant junction, mixed junction, stem-final mixed junction, vowel junction. We thus have to consider both syllable-level and word-level constituents.[5] These are summarised in Figure 1.

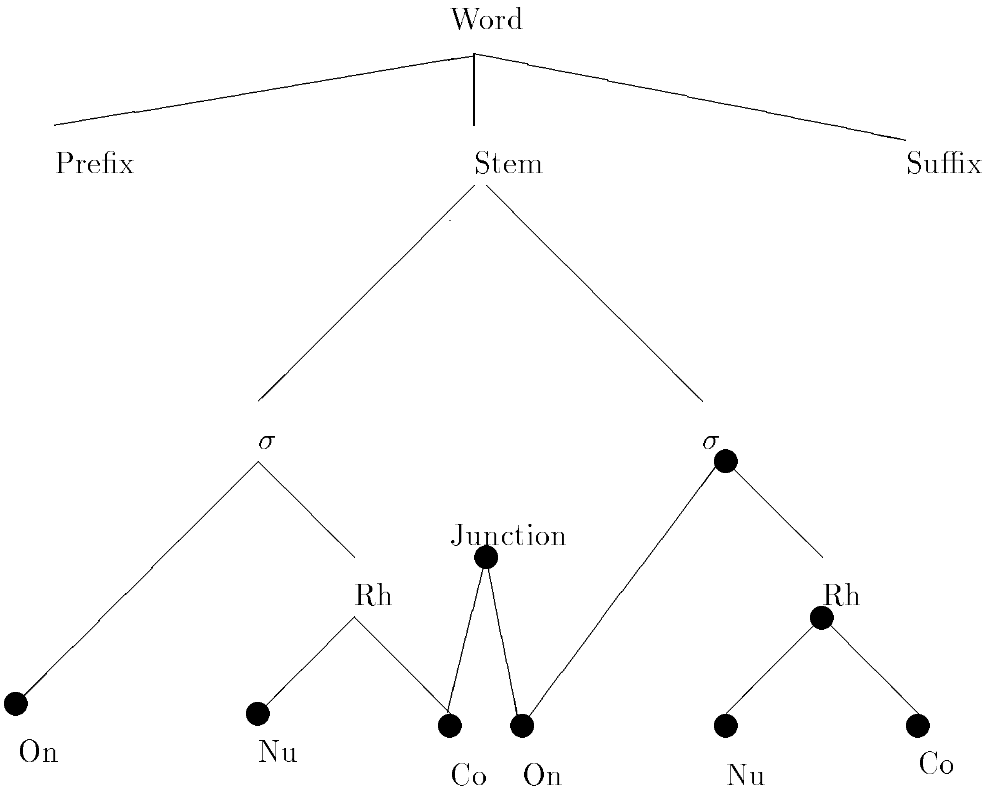

Figure 1: Domains for the application of speech variant rules

Large black dots mark possible domains for phonetic variant rules. The fact that

[5] We assume here are simple word schema consisting of a stem (bound or free) and optional affixes for derivates. More refined morphological constraints, such as the possible sequence of affixes, may be integrated later.

the occurrence of certain variants is tied to a particular syllable or word constituent can be advantageous in several respects:
When rules are used generatively, i.e. to produce variants from canonical forms, the constituent network serves as a filter: rules which do not take account of the constituent where a change takes place but which exclusively refer to the linear, segmental context will necessarily cause overgeneration. By contrast, if rules are arranged in a way that restricts their application to the relevant constituents, they do not generate ill-formed variants. Thus, a rule of the form

$$\text{g@n} \rightarrow \text{gn}$$

(i.e. "schwa is dropped between /g/ and /n/") would turn /g@ni:s@n/ into [gni:s@n], which is an impossible non-canonical variant. If schwa-deletion is restricted to word-final syllables, this problem does not arise.

When rules are used analytically, the input consists of a string of phonemes from a front-end recogniser. Information about syllable, morph or word boundaries is in this case not included in the input string. The fact that certain types of variants are limited to certain contituents can provide clues to syllable, stem or word boundaries.

Type of Change

In addition to the above criteria, rules can be distinguished as to the type of change they introduce. They may either affect entire segments (e.g. plosive epenthesis as in the case of [gants] for /gans/) or subsegmental features (e.g. place assimilation where only the place feature is changed but all other features (manner, voicing) are left intact, e.g. amfaN@n for anfaN@n). Segments can be deleted or inserted, features can be added, deleted or changed. Such a description is based on a segmental view of phonetic variation, which may be questioned in view of recent developments in phonological theory:

Multi-linear phonology has demonstrated the advantages of representing phonological entities in terms of tiers and association lines between the elements on those tiers. The operations described above would thus translate into manipulations of association lines (linking and delinking) or features, together with general principles of autosegmental phonology, such as the Obligatory Contour Principle (OCP)[6]. Our running example of voicing assimilation would be described in terms of multi-linear phonology as the deletion of the association line between

[6] The OCP states that identical adjacent elements on a given tier are automatically merged into one single element. This may be conceived of as a wellformedness filter on autosegmental representations rather than as an actual process

the /t/ sound and the [-voi] element[7]:

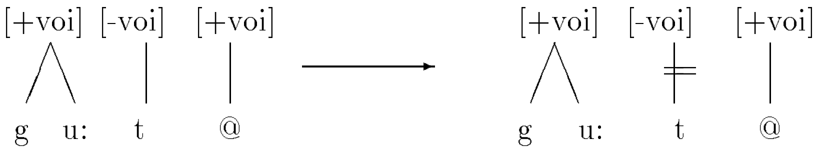

Figure 2: Deletion of association line in the case of voicing assimilation

Since two adjacent identical elements on a tier are merged into one by operation of the OCP, the resulting single [+voi] element automatically spans the /t/ sound, yielding the voiced apical plosive /d/:

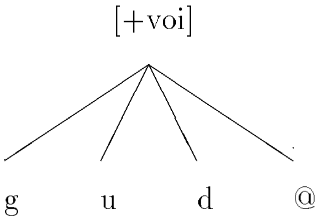

Figure 3: Operation of the OCP

In practice, however, an implementation of this analysis would turn out to be more complicated than needed from a computational point of view: The input and output strings both consist of segment symbols so that re-organisation of featural associations into discrete feature bundles would be required in any case. The internal featural representation of segments therefore resembles that of linear, SPE-style phonology (Chomsky & Halle (1968)).

The issue we then have to address is the choice of features. Our aim is to minimise the number of features used while maintaining the ability to correctly distinguish all phonemes and to model all speech variants. As for phonation, the features [+voice] and [-voice] are required. Manner features include [vowel], [nasal], [lateral], [continuant]. The feature [continuant] is sufficient to distinguish between fricatives ([+cont]) and plosives ([-cont]). Lip rounding in vowels is indicated by [round]. Place features are: [front], [central], [back], [high], [mid], [low], [labial], [coronal], [palato], [glottal], [uvular]. The feature [central] is used to distinguish lax vowels ([+central]) from tense vowels ([-central]). Evidence from

[7] Segment symbols are to be taken as abbreviations of non-relevant features.

phonetic feature recognition has shown that lax vowels can be defined by an overlap of the phonetic feature [central] with another place feature. By adopting this property for phonological feature definitions we do not have to include an extra feature [tense] or [lax]. Finally, the (actually suprasegmental) feature [long] has been included in the segmental feature list in order to be able to describe vowel shortening. All features are formally binary-valued. "Logical underspecification" has been employed in the sense that those features values which are predictable on the basis of phonetic (articulatory) constraints are not specified exhaustively. The feature [+vowel], for instance, implies negative values for all other manner features; therefore, the remaining manner features are left unspecified. Phonologically redundant specification was found to be necessary in those cases where the redundant feature was crucially required to model phonetic rules. Thus, although the voicedness of vowels is predictable, they have been redundantly specified as [+voice] to mark them as conditioning segments for voicing assimilation. As can be seen, the representation of segments and rules constitutes a compromise between phonological and phonetic considerations. The featural representations could be simplified and, at the same time, rendered more acceptable from a linguistic perspective if redundancy rules were given to fill in the phonologically redundant but phonetically required values.

4 Computational Aspects and Implementation

The question we now turn to is how speech variant rules should be implemented.

All speech variant rules are optional, and they can apply wherever a matching context can be found (or, in the case of context-free rules as defined above, wherever a relevant segment is present in the input string). Therefore, a canonical phonemic form may have multiple corresponding non-canonical forms. The simplest concept in order to generate all possible variants would be to take an unordered set of SPE-style rewrite rules and to use a simple pattern-matching procedure to determine whether a given rule can apply. Each time a context matched the input string, the appropriate change would be performed on the input. The resulting output string would then serve anew as the input to the entire set of rules, possibly triggering another change. This procedure would be iterated as long as rules could be found whose contexts match the input form. This approach, which may be termed the *iterative* application of rules, presents the following problems:

- First, as is the case with any unordered rewrite system, multiple identical output forms can be generated from the same input, depending on the order in which rules are applied. However, it would be preferable to generate not more than one token of each possible ouput form. For this reason, a checking procedure has to be included to determine whether a given variant has already been generated.

- Second, it may be the case that certain rules stand in a so-called feeding relation to each other, which means that the output of one rule is the input to another rule. This can be fatal if rules are contradictory. The rules of intersonorant voicing and initial devoicing in German illustrate this case: intersonorant voicing states that consonants may become voiced provided that the surrounding segments are sonorants. Initial devoicing, mostly operative in South German, effects the devoicing of syllable-initial coronal consonants. Here, one rule may produce an output form serving as input to the second rule and vice versa. An unordered set of rules, where each rule can apply an unlimited number of times, would, in this case, produce a non-terminating loop. This could be avoided by explicitly excepting rules from applying more than once. However, this procedure would be ad hoc and without any formal basis.

- Third, invalid or multiple output forms may be produced in cases where certain variant rules feed others. There is a rule in our corpus which states that vowels without primary stress may be reduced to schwa. The output forms created by this rules may thus contain a schwa which is then removed by the rule of schwa deletion, yielding unacceptable forms, e.g.
agEntIn → agEnt@n → agEntn
The final form is identical to the reduced form of *Agenten* but it is unacceptable as a variant of *Agentin*.

An alternative would be the so-called *sequential* application of rules: rules are extrinsically ordered; they are thereby barred from applying more than once. However, this equally entails some disadvantages:

- Rule ordering requires detailed knowledge about the interaction of rules in order to produce the correct rule sequence which generates all possible surface forms. Automatic learning of such a rule system is therefore rendered difficult if not impossible.

- The rule ordering cannot be inverted. If lexical (canonical) forms are to be recovered from surface forms, the rules may produce incorrect output forms or no output at all (cf. Kaplan & Kay (1994) for a more detailed criticism of this issue). Ordered rewrite rule systems thus are not declarative, contrary to our major requirement.

In general, rewrite rules make it difficult to assign probabilities to output forms. As will be seen below, it is desirable to pair output forms with their probabilities of occurrence in order to avoid overgeneration. In the case of ordered rules these probabilities would have to be calculated on a cumulative basis, being associated with derivations rather than with the individual rules themselves (cf. Cohen & Mercer (1975)); this increases computational load and inexplicitness of the system.

It was noted at an early date (Johnson (1972)) that phonological rewrite rules are equivalent in power to *finite-state transducers (FSTs)*[8] under the condition that they are barred from applying to their own output. FSTs are extensions of *finite-state automata (FSA)*. Informally, a FSA can be conceptualized as a device consisting of a read-head and a tape. The symbols written on the tape are read in one by one and the automaton advances one state. If all symbols have been read and the automaton ends up in a final state the string on the tape has been accepted. The difference between FSA and FSTs is that the latter have two tapes, one input and one output tape. As the read head moves along the tapes it reads a symbol from the input tape and writes a symbol on the output tape. A FST can thus act as a checking device (simply accepting symbols on both tapes), as a generator (generating an entire output form), or as a translator (translating the input to the output or vice versa).

Formally, a (two-tape) FST A is defined as a quintuple $\mathrm{A} = <\Phi, \Sigma_1, \Sigma_2, \delta, \lambda, \mathrm{S}>$, where

Φ is the set of states,
Σ_1 is the input alphabet ($\Sigma_1 \cap \Phi = \emptyset$),
Σ_2 is the output alphabet ($\Sigma_2 \cap \Phi = \emptyset$),
δ is the transition function $\delta : \Phi \text{ x } \Sigma_1 \rightarrow \Phi$,
λ is the output function $\lambda : \Phi \text{ x } \Sigma_1 \rightarrow \Sigma_2$, and
$\mathrm{S} \epsilon \Phi$ is the initial state.

The transducer thus defined is a *deterministic* transducer, i.e. both the transition and the output function produce singleton sets. In other words, given a specific state and a specific input symbol, the transducer may neither lead to more than one state nor allow multiple ouput symbols. In the case of non-deterministic transducers the transition and/or the output functions are set-valued: $\delta : \Phi \text{ x } \Sigma_1 \rightarrow \mathcal{P}(\Phi)$ and $\lambda : \Phi \text{ x } \Sigma_2 \rightarrow \mathcal{P}(\Sigma_2)$, where $\mathcal{P}$ is the power set of Φ and Σ_2, respectively.

Since rewrite rules which do not apply to their own output define regular relations and since regular relations correspond to FSTs, rewrite rules can equally

[8] For an introduction to formal languages and automata theory see e.g. Hopcroft & Ullman (1979).

be stated as FSTs. Ordered rewrite rules can accordingly be modelled by a *cascade* of FSTs, i.e. the output of one transducer serves as input to the next.

The fundamental equivalence between rewrite rules and transducers was rediscovered by Kaplan & Kay (1981)[9], who analysed the formal power of generative rewrite systems in an algebraic way: phonological rewrite rules define a very simple type of formal language, viz. regular relations. These are mappings between regular sets (expressions constructed from the elements of an alphabet by means of set union, concatenation or Kleene closure); they can also be thought of as regular sets whose expressions consist of pairs of symbols. Regular sets correspond to finite-state automata (FSA) whereas regular relations correspond to finite-state transducers (FSTs). They also demonstrated that a cascade of transducers could be converted into a single FST whose input and output correspond to the original input to, and the final output of, the cascade. This technique, called *composition*, produces a FST whose state set corresponds to the Cartesian product of the state sets of the basic transducers or to a subset thereof. Since regular relations are closed under composition the composite FST equally defines a regular relation.

Koskenniemi (1983) developed a slightly different computational model of linguistic rewrite rules, the so-called *two-level model*. Instead of composing a cascade of ordered transducers the relations between the input and the output level (linguistically speaking, the *lexical* and the *surface* level) can be stated directly in the form of two-level rules. These specify the correspondence pair and the lexical and surface contexts. Two-level rules are implemented as FSTs operating in parallel, each transducer defining a partial aspect of the entire change to the input form. This conjunctive linking of transducers ensures that a certain mapping is only accepted if no transducer blocks: The parallel transducers all scan the same string simultaneously, i.e. they all take the same transitions at the same time. Thus, a string is accepted if it is modeled by a path which is shared by all transducers. The accepted string belongs to the *intersection* of the two-level transducers.

No intermediate representations are involved in the two-level model. The linguist writing the rules thus does not have to think about rule ordering but can define correspondences between pairs of symbols directly. Another advantage is that the two-level system is declarative throughout; the direction of the mapping between lexical and surface level is irrelevant. Two-level rules fall into three categories:

(a) rules of the type $a : b \Rightarrow c : c\_\_d : d$,
i.e. a may be realised as b only in the context of both lexical and surface$c$ and d

[9] published as Kaplan & Kay (1994)

and nowhere else. These rules are optional.
(b) $a : b \Leftarrow c : c\_\_d : d$,
i.e. a must always be realised as b in the context $c : c\_\_d : c$, and possibly elsewhere.
(c) $a : b \Leftrightarrow c : c\_\_d : d$,
i.e. a must be always be realised as b in the context $c : c\_\_d : d$ and nowhere else. This rules combines the effects of rules (a) and (b).

Provided that the two-level transducers are ε-free (see below) they can equally be composed into a single transducer (Kartunnen (1983)) which is equivalent to a transducer derived by composition in the sense of Kay & Kaplan (see above). Koskenniemi's original model was extended further by allowing any type of regular expression, as well as disjunctions of sets and variables, to serve as contexts (cf. Black et al. (1987)).

Subsequently, the two-level model has been applied to multi-tiered, autosegmental descriptions, giving rise to "multi-level" models (Kay (1987)). By now, the use of finite-state devices in general, and two-level models in particular, has become a standard technique in computational linguistics; they have widely been used in morphology (e.g. Koskenniemi (1983), Kartunnen & Wittenburg (1983), Ritchie et al. (1987)), phonology (Carson (1988,1990,1992,1993)), Carson-Berndsen et al. (1989)) Carson-Berndsen & Gibbon (1992), Pulman & Hepple (1993), prosody (Gibbon (1987)), syntax (Kay (1983)), speech variants (Hoequist & Nolan (1991)) and part-of-speech tagging (Roche & Schabes (1994)). The most recent development in the domain of finite-state phonology is the work by Bird & Ellison (1992,1994), who suggest modelling both representations and rules in a single transducer, which results in so-called one-level phonology. However, their approach does as yet not capture changes which are not string-length-preserving. As we have seen, however, these operations frequently occur in continuous speech (elision, epenthesis), and we thus need a formal device which is able to capture these changes.

Determinism

A desirable feature of our implementation would be computational determinism. As it stands, our transducer contains non-determinism with respect to both the transduction function (i.e. given a state q and a word w, the transition function may yield a set of more than one states e.g. $\{q_1', q_2'\}$), and the output function (i.e. λ may yield an output set $\{w_1', w_2', ...w_n'\}$ for a configuration $\{q_i, w, q_f\}$, (where q_i is the initial and q_f is the final state, w is the input word)). Whereas every non-deterministic finite state automaton (FSA) can be converted into an equivalent deterministic FSA, this is not necessarily true for finite state *transducers*; only a subset of non-deterministic FSTs can be determinised. This subset consists of those FSTs which define rational functions on words, i.e. where non-determism is limited to the transition function and does not extend to the output function. However, our transducer defines a rational transduction, i.e. an input word w can be transduced to more than one output. Our FST thus cannot be completely determinised. This non-determinism, is however unavoidable since canonical forms usually have a number of corresponding non-canonical forms. This is true not only for the generation of non-canonical forms but also for the assignment of canonical representation to non-canonical forms. For a given speech variant, the transducer will in most cases output a number of possible canonical forms, of which only one is the correct, lexicalised form. This naturally leads to overgeneration relative to a given lexicon. We will return to this problem in section 7.

Declarativeness

Knowledge representations are declarative to the degree to which they favour the use of only one general procedural rule. This implies that a truly declarative representation is inherently non-directional with respect to its processing; in our case, this would mean that the phonetic knowledge base could be used both for generation and for recognition without having to make any changes to it.

Problems for a fully declarative approach are presented by deletion and epenthesis of segments. The reason for this is that a two-tape transducer as defined above requires that the number of input and output symbols be identical. If deletion of an element is to be modelled the element cannot simply be read in without producing any output on the second tape. Deleted or inserted segments might be mapped to the empty string ε. However, this presents formal problems for a two-level grammar:

As Kay & Kaplan (1994) have demonstrated, regular relations of unequal length are not closed under intersection. For instance, the intersection $R_1 \cup R_2$ of the regular relations $R_1 = \{< a^n, b^n c^* > | n \geq 0\}$ and $R_2 = \{< a^n, b^* c^n > | n \geq 0\}$ is $\{< a^n, b^n c^n > | n \geq 0\}$, whose range is the context-free language $b^n c^n$. On the other hand, the intersection of finite-state transducers always corresponds to a regular relation. In the case of ε-containing transducers, say T_1 and T_2, the relation defined by their intersection $T_1 \cup T_2$ is a subset of, or equal to, the intersection of the regular relations corresponding to the individual transducers $R_1 \cup R_2$. It is possible that a string exists whose corresponding relation is in $R_1 \cup R_2$ but which is not in $T_1 \cup T_2$: the string in question may simply be accepted by paths in the individual transducers which contain ε in different places and thus so not share a common sequence of transition labels. On the other hand, if the transducers are ε-free, i.e. if their corresponding strings have the same lengths, the relation defined by $T_1 \cup T_2$ is identical to $R_1 \cup R_2$; is must therefore be regular. Thus, in order to be able to rely on the finite-state property of two-level transducers is is necessary to have same-length strings, which excludes the use of ε. The solution is to use the null symbol *0* instead. 0 belongs to the characters of the alphabet and matches with deleted and inserted items. Thus, the strings to be paired do indeed have the same length and the corresponding relations are regular.

If we have a look at the two-level speech variant transducer, we see that there are practical problems with this approach:
Consider the case of schwa-deletion: in the case of the generation of a schwa-less surface form the schwa would be mapped to the null element: ?a:b@nt → ?a:b0nt. In order to reconstruct the canonical form in recognition, however, the input string would have to contain the null element in the appropriate position, which is impossible if the input comes from a phoneme recogniser. If the mapping were not as in Figure 4 but as in Figure 5 /n/ could be transduced to [@n] during recognition. This would be possible if the simple FST were replaced with a so-called two-tape *generalized sequential transducer (GST)*, which has the property that it need not necessarily be length-preserving. For any given symbol, the output can, apart from single symbols, consist of the empty symbol or a *string* of symbols. This property derives from the definition of the output function: $\lambda : \Phi \text{ x } \Sigma_1 \rightarrow \Sigma_2^*$, which states that symbols may be mapped to strings of symbols (including the empty string). Alternatively, this property can be brought about by dividing the states of the GST into two sets, printing states and scanning states. The transducer may then freely move from printing states to scanning states and vice versa. However, for the generation of the surface form two symbols would have to be read in by the transducer before the corresponding output symbol could be produced. This requires a computational memory and thus exceeds the formal power of a GST.

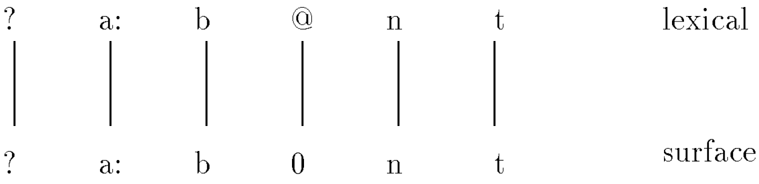

Figure 4

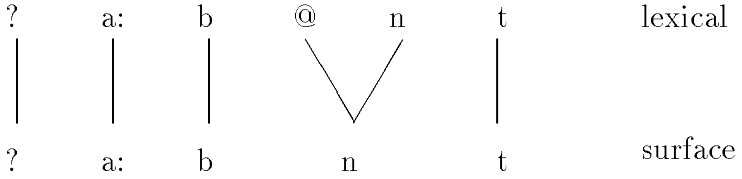

Figure 5

Clearly, null symbols must be introduced *before* the two-level mapping is carried out. As demonstrated by Kay & Kaplan (1994), this requirement is implicit in any two-level system: null symbols are "freely introduced" in lexical forms, which are then mapped to their corresponding surface forms or vice versa. Thus, the two-level system is technically a four-level system, with the two inner levels defining regular relations between lexical and surface forms. This method has also been adopted for the system under investigation; the core of the speech variant transducer is a two-level FST mediating between canonical and non-canonical forms which contain null symbols and are therefore of the same length. These are introduced by transducers mapping single symbols to themselves preceded or followed by null symbols, e.g. n → 0n (which is later mapped to @n). Thus, the outer transducers are non-declarative, directional GSTs encoding knowledge about possible insertion or deletion points of segments.

Let us now have a look at the procedure of building the speech variant transducer. For each rewrite rule, a corresponding two-level rule can be established which in turn can be stated as a FST (Figure 4). Since all speech variant rules are optional the corresponding two-level rules all belong to Koskenniemi's rule type A.

Rewrite rule:

[-voi] → [+voi] / [+voi] _____ [+voi]

Two-level rule: [-voi]:[+voi] ⇒ [+voi]:[+voi]_____ [+voi]:[+voi]

Transducer:

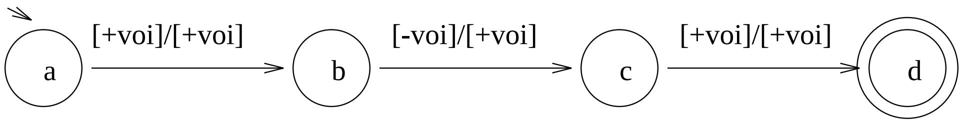

Figure 6: FST modelling voicing assimilation

After all two-level transducers have been built they can be composed in a large transducer similar to the one obtained by composition of cascaded transducers. The first step in this process is to modify transducers to the effect that they accept input strings whose length exceeds that of the string manipulated by a given rule (the so-called *factor*). This can be done by extending the transducer by adding the so-called "any" or "other" loops, i.e. paths which accept any symbol pair which is not further specified. The extended transducer transforms every factor in the input word into its corresponding output and accepts but leaves unchanged all other symbols in the input word. This can be done by adding an extra path which accepts all symbols not specified by the context (denoted by "="). Formally, "any" paths subsume all paths labelled with Σ-T_{qi}, where Σ is the set of all labels in the transducer alphabet, and T_{qi} is the set of labels on transitions leaving the state q_i.

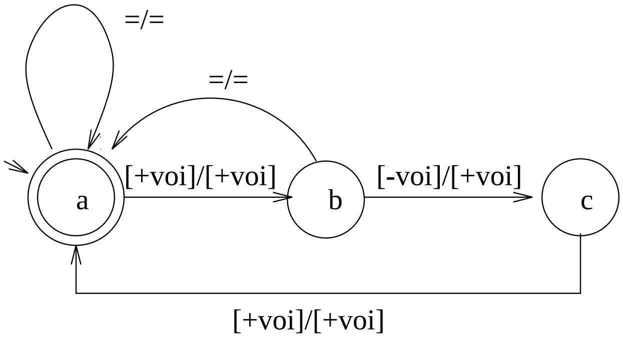

Figure 7: Extended FST modelling voicing assimilation

The second step is to group transducers together according to their domain of application and to compose the resulting set of FSTs. The comparatively small onset network can be used as an example: the rules operative in the onset are affricate reduction (A), glottal stop dropping (B) and initial devoicing (C) (see section 6). Additionally, none of these rules may be applied; in this case the onset may contain up to three consonants (D). These possibilities are expressed by the following transducers:

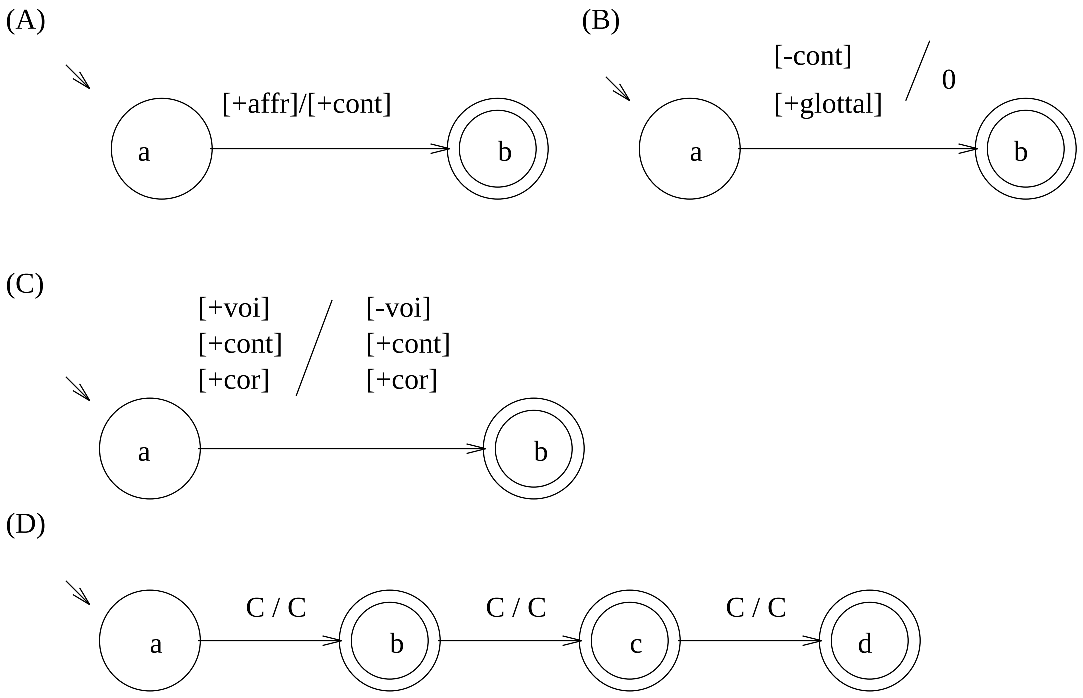

Figure 8: Onset transducers

The composite onset transducer (Figure 9) in this case simply combines the initial and final states of all transducers into a single initial state and a single final state and conflates the transitions of (A), (B) and (C).

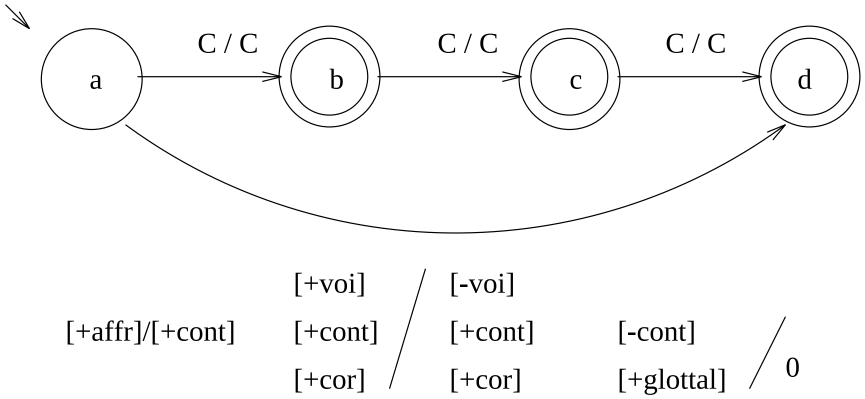

Figure 9: Composite onset transducer

Finally, all constituent networks are joined together to form one large transducer network modelling the structure of a word together with possible variants located in their domains of application. The extension procedure can be additionally be used to specify constraints on syllable structure, which is required when the syllable network is used simultaneously as a wellformedness checking device on canonical input forms: in this case, not *any* segment or number of segments is allowed to precede and to follow the context but only a certain number of certain types of segments, as determined by the phonotactics of the constituent in question (see Carson-Berndsen et al. (1989), Carson-Berndsen (1990)). Thus, in Figure 8 all arcs in (D) are labelled with the category C, taken as an abbreviation of the set of consonants, since only consonants are allowed in these positions.

This combination of finite-state devices serving different purposes (stating wellformedness constraints v. characterising legal mappings) represents a novel addition to existing two-level techniques: two-level systems alone do not define well-formedness constraints on input forms; they merely constrain the *mapping* between two levels of representation. However, if well-formedness constraints can be stated in terms of a finite-state automaton, the FST derived from this FSA can be composed with the two-level FST. The closure property is preserved under composition exactly as in the case of composition of specific two-level FSTs.

Implementation

For test purposes the speech variant transducer has been implemented in a a PROLOG program for the generation of possible speech variants of German. The following steps are involved: First, a wordlist containing canonical transcriptions together with stress and syllable information is read in, and segment symbols are converted into feature lists according to the specifications given in the conversion tables in the appendix.

Second, the individual entries are processed by the speech variant transducer. This transducer contains as its knowledge base a large transition network, which in turn encompasses several subnetworks corresponding to the domains of application of rules (nucleus, coda, etc.). Since networks may be embedded (e.g. the syllable network comprises an onset network and a rhyme network, which in turn consists of a nucleus network and a coda network) the speech variant transducer has to be enriched with a memory. Thus, the resulting device is actually a pushdown transducer, not a FST. This additional property, however, has been introduced entirely for reasons of notational convenience and linguistic generalisation. From a computational perspective, its power is not needed: the language to be processed is still finite-state and not context-free, as might be expected. The formal finite-state property could be preserved by replacing the arcs labelled with subnetwork symbols with the corresponding subnetwork arcs themselves. However, considering the size and the complexity of the transition network, the

possibility of being able to specify embedded structures seems to be the better solution. The two-level rules correspond to arcs in the network labelled with legal pairs of canonical and non-canonical segments or segment classes; the latter are implemented as underspecified feature lists. The input feature specifications (lists of complex terms) are matched against the descriptions on the arcs by means of term unification. During transition of the network (depth-first search), the input forms are both parsed into their constituents and changed into their corresponding non-canonical forms. Generation of a speech variant terminates sucessfully when a final state has been reached and no input is left. Backtracking ensures that all possible variants are generated.
Finally, any resulting non-canonical output form is reconverted to segment symbols and is stored in a file. Examples of output files are given in the appendix.

5 Practical Application

We will now illustrate possible applications of the word-level speech variant transducer, in particular its integration into a knowledge-based word recogniser developed at the University of Bielefeld. First, however, let us consider its generative use:

Generation
As described above, possible speech variants can be generated by an exhaustive search through the transducer network. In this way all valid paths through the network are transduced, where each path corresponds to a possible variant.

The output may then be combined with, say, a HMM recognizer for the task of word-level recognition. If the canonical transcription of the utterance to be labelled is known, the transcription of each word can serve as the input to the speech variant transducer. The final output will be a lattice of transcriptions, including canonical as well as non-canonical forms. The Viterbi algorithm can be used in order to detect the most likely path through this lattice; further re-estimation is then performed on the forms picked out by this path to improve segment boundaries.

In order to avoid excessive generation of variants, it is useful to provide syllable boundary markers and stress markers in the input string. Since the speech variant transducer can output syllabification hypotheses, multiple copies of a single variant are produced due to multiple possibilities of placing the syllable boundaries. This is true for stress markers as well. The transducer contains different subnets for fully stressed vowels and vowels with secondary or no stress. If information about stress is unavailable, all possible output forms will be generated. Thus, to limit the number of speech variants it is advisable to supply this information.[10] Using this transducer speech variants were generated for all entries in the VERBMOBIL demonstrator wordlist (U Bielefeld). The result was used for the training of a HMM-based recogniser in VM TP 15.3 (*Phonologisch informierte akustische Analyse*, U Hamburg). First, forced alignment of was carried out between speech data and a set of alternative phonemic transcriptions containing speech variants; in each case the transcription sequence with the maximum a-posteriori probability was chosen. In a second step this material was used to re-train a HMM-based recogniser, which resulted in a slight increase in recognition rate compared to a recogniser trained on exclusively canonical material.

Recognition

As we mentioned in the introduction, the speech variant transducer has been established for a specific purpose, viz. to enrich a knowledge-based word recogniser (*BELLEx3*, cf. Gibbon et al. (1992), Althoff et al. (1995)). This recogniser has the following architecture: At the bottom level, acoustic-phonetic events (such as front vowel, nasality, fricative noise, etc.) are extracted from the speech signal by a statistical event detector (*HEAP*, see Hübener (1993), Hübener & Carson-Berndsen (1994)). These are then combined to phonologically relevant features on the basis of a temporal event logic specifying permissible overlap and precedence relations between acoustic-phonetic events. The output of this module consists of (possibly underspecified) feature bundles, the combination of which is further constrained by a syllable parser (*SILPA*, cf. Carson-Berndsen (1993), Hübener & Carson-Berndsen (1994)). This parser contains a complete phonotactic network describing all possible German syllable structures (Carson-Berndsen (1992)). Phoneme hypotheses are passed on to a higher level morphotactic parser provided that they conform to the phonotactic contraints imposed by the network. Thus, at the onset of a syllable at most three consonants are allowed, the combination of which is additionally dependent on manner and place features. These properties limit the number of permissible segments at certain stages in the recognition process and allows detailed top-down predictions to be made.

Finally, the syllable and segment class hypotheses are mapped onto lexical representations by an Earley-based morphoprosodic parser in finite-state transducer form (*MORPROPA*). The output from this module consists of a lattice of underspecified word hypotheses which are passed on to a syntactic parser. The morphoprosodic parser performs three actions; PREDICT, during which a specific top-down hypothesis is made, SCAN (scanning of the input) and COM-

[10] It is of course possible to automatise these steps and to connect the speech variant transducer to syllabification and/or a stress assignment algorithm which are run on the input data *before* the generation of variants takes place.

PLETE (move along the transducer path whenever the input symbol and the hypothesised symbol match). The speech variant knowledge base is integrated into this parser in the following way: an attempt is made at matching pairs of input/output symbols (as specified in the individual transducers corresponding to speech variant rules) with PREDICT and SCAN symbols. In case of a mismatch between PREDICT and SCAN symbol due to the presence of a speech variant a further SCAN symbol may be provided by the transducer if some rule in our rule corpus is applicable: The input symbol in this case matches with the non-canonical member of the input/output pair of the transducer. The canonical member of this pair is then matched with the PREDICT symbol. If this match succeeds the morphoprosodic parser performs a COMPLETE operation.

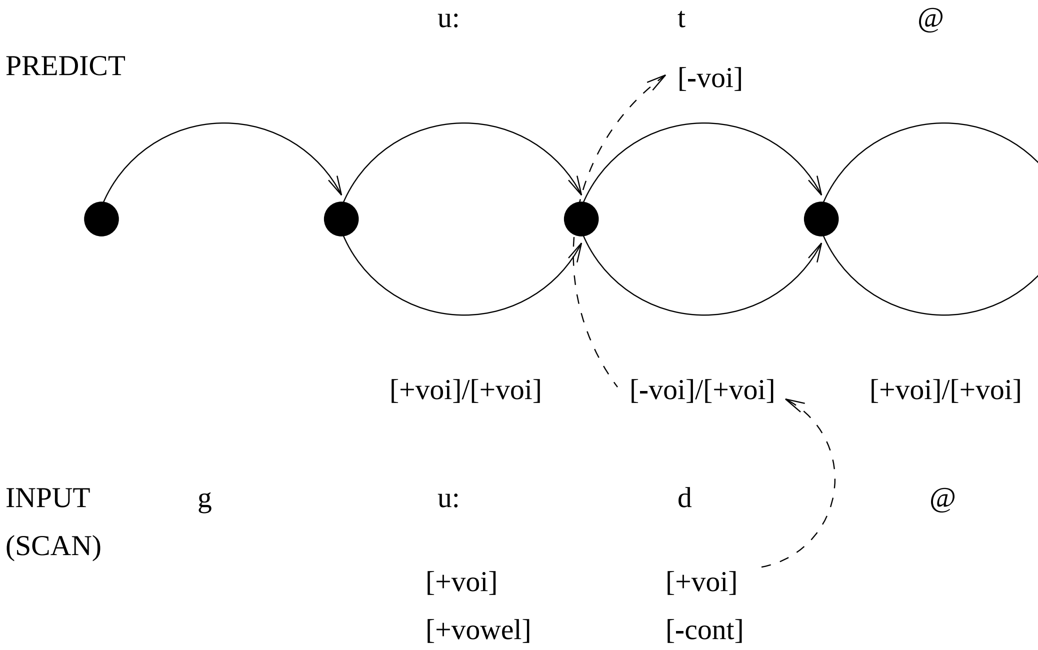

Figure 10: Integration of speech variant rules in a morphoprosodic parser

Due to the event-based approach, speech variants which result from timing shifts in the realisation of articulatory gestures can in principle be resolved by the syllable parser. Consider the case of nasal assimilation in the syllable coda, e.g. fYnf → fYmf. Here the labial event overlaps both with a nasal event and with a fricative event. When this overlap relation is detected by the syllable parser in the coda, two alternative segment hypotheses (/m/ and /n/), or an underspecified feature bundle which simply states that a nasal is present, can be passed on. There is good evidence that all reduction phenomena in German can be modelled by shifts in the timing relations between events: in the context of speech synthesis based on the paradigm of articulatory phonology, all segmental reduction phenomena cited by Kohler (1990) have been shown to be producible by specifications of articulatory gestures and appropriate timing relations between them (Kröger (1993)). However, not all speech variants can be confined to the syllable constituent. As we have discussed above, there are variants which depend on syllable junctions or on the word structure. Information about syllable and word structure therefore has to be combined in order to account for *all* variants. This can be done by using the speech variant word transducer as a module mediating between segmental input and the lexicon in combination with the morphotactic parser, as described above. Another possibility would be to consider syllable-level and word-level rules separately, integrating the former set into the syllable-parser and the latter into the morphotactic parser. Practical evaluation of these different concepts will have to show which one is preferable.

6 The Speech Variant Knowledge Base

In this section we will describe the actual rules. They are grouped according to their domain of application as specified in the structure of the transducer network. Assigning rules to constituents entails that no boundary symbols have to be included in the rules. On the other hand, the same rules may need to be specified more than once in the transducer network, viz. in every constituent network where the rule is operative. Here, rules will be cross-referenced where appropriate. The format of the rules is that of Koskenniemi's type (A) rules as described in section 4. Only those features which are crucially relevant for the application of rules are given; variables like [α place] are to be taken to represent any feature-value pair relevant for the category in question (cf. the informal grouping of features in section 2). No context is given for context-free rules; in the case of rules where only one side of the context is relevant the other side is filled by the "any" pair =/=.

Stem-initial onset

Stem-initially, two phenomena may take place: initial devoicing and the reduction of affricates. Initial devoicing states that (stem-)initial voiced coronal fricatives may become voiceless. Although we have found a number of other consonants undergoing initial devoicing, a relaxation of the constraints [+coronal] and [+cont] would lead to highly unlikely variants and thereby to overgeneration. The rule is context-free. Furthermore, is is mostly limited to South-German variants.

(1) Stem-initial devoicing

$$\begin{bmatrix}\text{+voi}\\ \text{+cont}\\ \text{+cor}\end{bmatrix} : \begin{bmatrix}\text{-voi}\\ \text{+cont}\\ \text{+cor}\end{bmatrix}$$

Examples: zamsta:k → samsta:k, g@za:kt → g@sa:kt

Affricates can be reduced to simple fricatives stem-initially. Non-initial reduction,e.g. [fI6se:n] for /fI6tse:n/, is less likely.

(2) Reduction of affricates

[+affr]:[+cont]

Examples: pfEnd@n → fEnd@n, tsy:g@ → sy:g@

Onset

At the onset of any syllable a glottal stop can be dropped if present in the standard pronunciation.

(3) Glottal Stop Dropping

? : 0

Example: ?anfaN@n → anfaN@n

Nucleus

The nucleus is the domain of vowel changes, such as shortening, laxing or the reduction of diphthongs. These changes are the only ones dependent on the prosodic context. As far as this context is concerned, we have found out that the most important feature is the presence vs. absence of primary stress: only vowels without primary stress can be reduced to schwa. Laxing and schwa-reduction, by contrast, may apply to both to vowels with secondary stress and to unstressed vowels. Shortening can apply to all vowels. The degree of stress (primary v. secondary) thus only plays a minor role; it may affect the probability of the occurrence of a certain vowel change, but in most cases it does not constitute a necessary condition. Since different subnetworks exist for fully stressed vowels and for those with secondary or no stress, it is unnecessary to include stress marks in the rules.

(4) Shortening

$$\begin{bmatrix} +\text{vowel} \\ +\text{long} \\ \alpha\ \text{place} \end{bmatrix} : \begin{bmatrix} +\text{vowel} \\ \text{-long} \\ \alpha\ \text{place} \end{bmatrix}$$

Example: ba:nho:f → banhof

(5) Laxing

$$\begin{bmatrix} +\text{vowel} \\ \text{-central} \\ \alpha\ \text{place} \end{bmatrix} : \begin{bmatrix} +\text{vowel} \\ +\text{central} \\ \alpha\ \text{place} \end{bmatrix}$$

Example: ba:nho:f → ba:nhOf

We need a separate rule to account for laxing from /e:/ to /E/ since the latter is not defined by the feature [central] in our system. Instead, it is distiguished from its counterpart /E:/ by absence or presence of the feature [long].

$$\begin{bmatrix} +\text{vowel} \\ +\text{long} \\ +\text{mid} \\ +\text{front} \\ \text{-round} \end{bmatrix} : \begin{bmatrix} +\text{vowel} \\ \text{-long} \\ +\text{low} \\ +\text{front} \\ \text{-round} \end{bmatrix}$$

Examples: fYnftse:n → fYnftsEn

The next rule is primarily indicative of North German speech but cannot be limited to this dialect region.

(6) E:/e: Alternation

$$\begin{bmatrix} +\text{vowel} \\ +\text{long} \\ +\text{low} \\ +\text{front} \\ \text{-round} \end{bmatrix} : \begin{bmatrix} +\text{vowel} \\ +\text{long} \\ +\text{mid} \\ +\text{front} \\ \text{-round} \end{bmatrix}$$

Example: mE:tC@n → me:tC@n

Rule 7 equally describes a slightly dialectal feature. However, it cannot strictly be localised, as speakers from regions as diverse as the Rhenish area and the Baltic Sea coastal area exhibit this feature. In view of this wide distribution it seemed justified to include this rule in our corpus. It applies to both fully stressed and not fully stressed vowels.

(7) I-Rounding

$$\begin{bmatrix} +\text{vowel} \\ +\text{high} \\ +\text{front} \\ +\text{central} \\ \text{-round} \end{bmatrix} : \begin{bmatrix} +\text{vowel} \\ +\text{high} \\ +\text{front} \\ +\text{central} \\ +\text{round} \end{bmatrix} \Rightarrow=:= \_\_ \begin{bmatrix} +\text{vowel} \\ +\text{low} \\ +\text{central} \end{bmatrix} : \begin{bmatrix} +\text{vowel} \\ +\text{low} \\ +\text{central} \end{bmatrix}$$

Example: tsI6ka: → tsY6ka:

As mentioned above, any vowel without primary stress can be reduced to schwa. Naturally, the tendency for schwa-reduction is stronger for unstressed vowels than for vowels with secondary stress; although the above rule may cause overgeneration it would be too restrictive to limit it to vowels with secondary stress.

(8) Schwa-Reduction

$$\begin{bmatrix} +\text{vowel} \\ \text{-central} \\ \alpha\ \text{place} \end{bmatrix} : \begin{bmatrix} +\text{vowel} \\ +\text{mid} \\ +\text{central} \end{bmatrix}$$

Example: de:tsEmb6 → d@tsEmb6

We now turn to the reduction of diphthongs. This includes the laxing and the shortening of the first element (/mi:6/ → [mi6], /mi:6/ → [mI6]) and the dropping of /6/ in the case of falling diphthongs. Unlike in the case of single vowels, laxing also applies to fully stressed diphthongs. Diphthongs without primary stress can be reduced to schwa. Rising diphthongs, i.e. /aI/, /OY/, /aU/, sometimes are reduced to single vowels, especially in function words (thus for instance [?an@] for /?aIn@/). Nevertheless, this phenomenon is too sporadic to be included as a rule.

The laxing of the first element of diphthongs can be described by Rule 5 since diphthongs are considered sequences of two vowels in our system. Similarly, the shortening of the first element is described by Rule 4.

(9) Diphthong Reduction: 6-dropping

$$\begin{bmatrix} \text{+vowel} \\ \text{+low} \\ \text{+central} \end{bmatrix} : 0 \Rightarrow [\text{+vowel}]{:}[\text{+vowel}] \;\_\_\; =:=$$

Example: hambU6k → hambUk, fI6tse:n → fItse:n

(10) Diphthong Reduction: reduction to schwa

$$\begin{bmatrix} \text{+vowel} \\ \alpha \text{ place} \end{bmatrix} : 0 \Rightarrow =:= \;\_\_\; \begin{bmatrix} \text{+vowel} \\ \text{+low} \\ \text{+central} \end{bmatrix} : \begin{bmatrix} \text{+vowel} \\ \text{+mid} \\ \text{+central} \end{bmatrix}$$

Example: hambU6k → hamb@k

The diphthongs /E6/, /E:6/, /e6/ and /e:6/ can all be reduced to [6] in case they do not carry primary stress.

(11) Diphthong Reduction: reduction to a-schwa

$$\begin{bmatrix} \text{+vowel} \\ \text{-high} \\ \text{+front} \\ \text{-central} \\ \text{-round} \end{bmatrix} : 0 \Rightarrow =:= \;\_\_\; \begin{bmatrix} \text{+vowel} \\ \text{+low} \\ \text{+central} \end{bmatrix} : \begin{bmatrix} \text{+vowel} \\ \text{+central} \\ \text{+low} \end{bmatrix}$$

Example: E6klE:r@n → 6klE:r@n

/E:/ and /E/ as first elements in a diphthong can be altered to /e:/ and /e/, respectively. This applies to diphthongs both with and without primary stress. Rule 6 is sufficient to account for this change.

Example: E6klE:r@n → e6klE:r@n

Coda

In the coda assimilation, epenthesis and elision take place. Nasal assimilation states that an alveolar nasal adopts the place features of the following consonant. Since the velar nasal is limited in its distribution to the coda, the rule might equally be limited to coronal nasals preceding labial consonants. However, we can use the general rule of a coronal nasal and the following consonant sharing the place feature as a wellformedness constraint on input forms. Moreover, the same rule can be used below for nasal assimilation across syllable boundaries. Further variants which can be assigned to the coda constituent are glottalisation of plosives and r-alternation.

(12) Nasal assimilation

$$\begin{bmatrix} \text{+nas} \\ \text{+cor} \end{bmatrix} : \begin{bmatrix} \text{+nas} \\ \alpha \text{ place} \end{bmatrix} \Rightarrow=:= \_\_ \begin{matrix} \text{C} \\ [\alpha \text{ place}] \end{matrix} : \begin{matrix} \text{C} \\ [\alpha \text{ place}] \end{matrix}$$

Examples: fYnf → fYmf

Between nasals and following fricatives, plosives can be inserted which share the place feature of the nasal:

(13) Plosive epenthesis

$$0 : \begin{bmatrix} \text{-cont} \\ \alpha \text{ place} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{+nas} \\ \alpha \text{ place} \end{bmatrix} : \begin{bmatrix} \text{+nas} \\ \alpha \text{ place} \end{bmatrix} \_\_ \begin{bmatrix} \text{-voi} \\ \text{+cont} \end{bmatrix} : \begin{bmatrix} \text{-voi} \\ \text{+cont} \end{bmatrix}$$

Examples: gans → gants, rINs → rINks

The opposite phenomenon also takes place: if the canonical form contains the sequence nasal-plosive-fricative the plosive can be deleted.

(14) Plosive Deletion

[-cont]: 0 ⇒ [+nas]:[+nas] __ [+cont]:[+cont]

Examples: ko:blEnts → ko:blEns, lINks → lINs

Syllable-final stops can be glottalised.[11]

(15) Glottalisation of plosives

$$\begin{bmatrix} \text{-cont} \\ \alpha \text{ place} \end{bmatrix} : \begin{bmatrix} \text{-cont} \\ \text{+glottal} \end{bmatrix}$$

[11] In most cases the term "glottalisation" refers to sounds produced with a closed glottis but nevertheless preserving their supralaryngeal features. Since we do not take into account any such subphonemic variants we use "glottalisation" for the replacement of consonants with a glottal stop, i.e. the complete effacement of supralaryngeal features.

Example: dO6tmUnt → dO6tmUn?, fraNkfU6t → fraNkfU6?

Our final coda rule describes the realisation of /r/ preceding a consonant. We take the canonical transcription of these sequences to be /6/C, e.g. *Fahrt* = /fa:6t/. However, /r/ may actually be realised by a uvular fricative constriction, i.e. [fa:rt], or, in the case of dialectal pronunciation, by an alveolar tap or trill. In the coda velar fricative weakening takes place: The velar fricative /x/ can be weakened to /h/. This is brought about by laxing of the velar constriction, leaving only the fricative gesture but not the place feature. Although this kind of weakening may apply to the palatal fricative /C/ as well, there are not enough cases in our data to warrant the extension of the rule given below.

(16) R-alternation

$$\left[\begin{array}{l}\text{+vowel}\\ \text{+low}\\ \text{+central}\end{array}\right] : \left[\begin{array}{l}\text{+voi}\\ \text{+cont}\\ \text{+back}\end{array}\right] \Rightarrow [\text{+vowel}]:[\text{+vowel}] \,\underline{}\, \text{C:C}$$

(17) Velar fricative weakening

$$\left[\begin{array}{l}\text{-voi}\\ \text{+cont}\\ \text{+back}\end{array}\right] : \left[\begin{array}{l}\text{-voi}\\ \text{+cont}\\ \text{+glottal}\end{array}\right] \Rightarrow \text{=:=} \,\underline{}\, \text{C:C}$$

Example: maxt → maht

Rhyme

As far as *regular* l-vocalisations are concerned, we have found a single case: /l/ becomes a palatal lax vowel (/I/) between a palatal vowel and a palatal fricative.

(18) L-vocalisation

$$[\text{+lat}] : \left[\begin{array}{l}\text{+vowel}\\ \text{+high}\\ \text{+front}\\ \text{+central}\\ \text{-round}\end{array}\right] \Rightarrow \left[\begin{array}{l}\text{+vowel}\\ \text{-low}\\ \text{+front}\end{array}\right] : \left[\begin{array}{l}\text{+vowel}\\ \text{-low}\\ \text{+front}\end{array}\right] \,\underline{}\, \left[\begin{array}{l}\text{+cont}\\ \text{+front}\end{array}\right] : \left[\begin{array}{l}\text{+cont}\\ \text{+front}\end{array}\right]$$

Example: vElC → vEIC

Stem-final syllable

The stem-final syllable is the domain for schwa-deletion and all processes related to it.

(19) Schwa-Deletion

$$\begin{bmatrix} +\text{voi} \\ +\text{vowel} \\ +\text{mid} \\ +\text{central} \end{bmatrix} : 0 \Rightarrow \text{C:C} \;\_\_\; \{[+\text{nas}]:[+\text{nas}] \mid [+\text{lat}]:[+\text{lat}]\}$$

Examples: trEf@n → trEfn, l9f@l → l9fl

If schwa-deletion does take place, the final nasal, if coronal, may assimilate to the preceding consonant in its place of articulation features (Rule 20). In the case of inflected determiners, such as /?aIn@m/, a coronal nasal preceding schwa assimilates to the final nasal (Rule 21).

(20) Schwa-deletion and progressive place assimilation

$$\begin{bmatrix} +\text{vowel} \\ +\text{mid} \\ +\text{central} \end{bmatrix} : 0 \Rightarrow \begin{matrix} \text{C} \\ [\alpha \text{ place}] \end{matrix} : \begin{matrix} \text{C} \\ [\alpha \text{ place}] \end{matrix} \,\text{—}\, \begin{bmatrix} +\text{nas} \\ +\text{cor} \end{bmatrix} : \begin{bmatrix} +\text{nas} \\ \alpha \text{ place} \end{bmatrix}$$

Example: glaUb@n → glaUbm, li:g@n → li:gN

(21) Schwa-deletion and regressive place assimilation

$$\begin{bmatrix} +\text{vowel} \\ +\text{central} \\ +\text{mid} \end{bmatrix} : 0 \Rightarrow \begin{bmatrix} +\text{nas} \\ +\text{cor} \end{bmatrix} : \begin{bmatrix} +\text{nas} \\ +\text{lab} \end{bmatrix} \text{—} \begin{bmatrix} +\text{nas} \\ +\text{lab} \end{bmatrix} : \begin{bmatrix} +\text{nas} \\ +\text{lab} \end{bmatrix}$$

Example: ?aIn@m → ?aImm

If schwa is deleted and the nasal assimilates, a voiced plosive preceding schwa in the canonical form can be deleted or assimilated to the nasal. Similarly, a nasal preceding schwa in the lexical form may disappear as well, leaving only the final nasal. The decision of whether two nasals or one nasal are present is dependent on the perception of the hearer.

(22) Schwa-deletion and consonant deletion

$$\text{(a)}\begin{bmatrix}\text{+vowel}\\ \text{+mid}\\ \text{+central}\end{bmatrix} : 0 \Rightarrow \begin{matrix}\text{+voi}\\ \text{-cont}\\ [\alpha\ \text{place}]\end{matrix} : 0 \,\_\_\, \begin{bmatrix}\text{+nas}\\ \text{+cor}\end{bmatrix} : \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix}$$

Example: glaUb@n → glaUm

$$\text{(b)}\begin{bmatrix}\text{+vowel}\\ \text{+mid}\\ \text{-central}\end{bmatrix} : 0 \Rightarrow \begin{bmatrix}\text{+nas}\\ \text{+cor}\end{bmatrix} : 0\_\_ \begin{bmatrix}\text{+nas}\\ \text{+lab}\end{bmatrix} : \begin{bmatrix}\text{+nas}\\ \text{+lab}\end{bmatrix}$$

Example: ?aIn@m → ?aIm

(23) Schwa-deletion and consonant assimilation

$$\begin{bmatrix}\text{+vowel}\\ \text{+central}\\ \text{+mid}\end{bmatrix} : 0 \Rightarrow \begin{bmatrix}\text{-cont}\\ \alpha\ \text{place}\end{bmatrix} : \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix} \_\_ \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix} : \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix}$$

Examples: glaUb@n → glaUmm

Finally, if the consonant preceding the schwa in the canonical form is a voiceless stop, glottalisation may take place (cf. Rule 15).

Vowel Junction

In our terms, a vowel junction is a syllable junction where a single consonant is enclosed by two vowels. This configuration may trigger the weakening of the intervocalic consonant, which may surface as either voicing assimilation or laxing of the constrictory gesture in the consonant or both. Laxing of the constriction produces approximants or even fricatives. Thus, the labial voiced plosive /b/ can result in a bilabial voiced fricative, which is not included in the segment inventory of German. However, if fricativisation is very strong, this variant is sometimes transcribed as /v/ in label files (Rule 25). Velar plosives may be fricativised; these variants are regularly found in certain dialects, such as Westphalian. There is no symbol available in German to represent voiced velar fricatives. However, when preceded by a [+high] or [+mid] vowel, the fricative can be palatalised, yielding /j/. This is somewhat reminiscent of the Berlin or Rhenish dialect, but it may equally arise in casual standard speech (Rule 26).

(24) Voicing Assimilation

$$\begin{matrix}\text{C}\\ \text{[-voi]}\end{matrix} : \begin{matrix}\text{C}\\ \text{[+voi]}\end{matrix} \Rightarrow \text{[+voi]:[+voi]} \_\_ \text{[+voi]:[+voi]}$$

Example: SpE:t@st@ns → SpE:d@st@ns, QUnt6brIN@n → QUnd6brIN@n

(25) Laxing of Constriction: labials

$$\begin{bmatrix}\text{+voi}\\ \text{-cont}\\ \text{+lab}\end{bmatrix} : \begin{bmatrix}\text{+voi}\\ \text{+cont}\\ \text{+lab}\end{bmatrix} \Rightarrow \text{[+vowel]:[+vowel]} \_\_ \text{[+vowel]:[+vowel]}$$

Example: ?ab6 → ?av6, detsEmb6 → detsEmv6

(26) Laxing of constriction: velars

$$\begin{bmatrix}\text{+voi}\\ \text{-cont}\\ \text{+back}\end{bmatrix} : \begin{bmatrix}\text{+voi}\\ \text{+cont}\\ \text{+front}\end{bmatrix} \Rightarrow \text{[+vowel]:[+vowel]} \_\_ \text{[+vowel]:[+vowel]}$$

Example: li:g@va:g@n → li:j@va:g@n

Consonant Junction

A consonant junction is a syllable junction where two or more consonants come together. In this domain we find voicing assimilation, manner assimilation, plosive deletion, l-vocalisation, consonant cluster reduction, velar fricative weakening, and glottalisation of plosives.

In homorganic nasal-plosive sequences the plosive can assimilate to the following nasal, yielding a nasal-nasal sequence. Again, the sequence can be simplified to a single nasal. Furthermore, plosives can be deleted in the context of a nasal to the left and a fricative to the right.

(27) Cross-syllable plosive assimilation/deletion

(a) assimilation

$$\begin{bmatrix}\text{-cont}\\ \alpha\ \text{place}\end{bmatrix} : \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix} : \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix} \_\_ \text{=:=}$$

Example: hambU6k → hammU6k

(b) deletion

$$\begin{bmatrix}\text{-cont}\\ \alpha\ \text{place}\end{bmatrix} : 0 \Rightarrow \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix} : \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix} \_\_ =:=$$

Example: hambU6k → hamU6k

(c) deletion before fricatives

$$\begin{bmatrix}\text{-cont}\\ \alpha\ \text{place}\end{bmatrix} : 0 \Rightarrow \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix} : \begin{bmatrix}\text{+nas}\\ \alpha\ \text{place}\end{bmatrix} \_\_ \text{[+cont]:[+cont]}$$

Example: fraNkfU6t → fraNfU6t

Voicing assimilation takes the same form as Rule 24. The rule for l-vocalisation is identical to the one given above for l-vocalisation in the rhyme (18). For velar fricative weakening and glottalisation of plosives see (17) and (15).

Many kinds of consonant cluster simplification may occur in connected speech. One regular type of reduction is is the simplification of heterosyllabic /st/ clusters, where /t/ is deleted.

(28) Cluster simplification

$$\begin{bmatrix}\text{-voi}\\ \text{-cont}\\ \text{+cor}\end{bmatrix} : 0 \Rightarrow \begin{bmatrix}\text{-voi}\\ \text{+cont}\\ \text{+cor}\end{bmatrix} : \begin{bmatrix}\text{-voi}\\ \text{+cont}\\ \text{+cor}\end{bmatrix} \_\_ =:=$$

Examples: nE:Cst@ → nE:Cs@, gr2:st@n → gr2:s@n

Mixed Junction

A mixed junction is a syllable junction where a vowel is followed by a consonant. In practice, this consonant is always /r/.

Schwa-Lowering indicates that /@/ becomes /6/ before /r/. The lowering gesture which has to be performed by the tongue in order to articulate the /r/ sound (which, in Standard German, has a uvular quality) is anticipated in the preceding vowel. Lowering of a central schwa yields the a-schwa.

(29) Schwa-Lowering

$$\begin{bmatrix}\text{+vowel}\\ \text{+mid}\\ \text{+central}\end{bmatrix} : \begin{bmatrix}\text{+vowel}\\ \text{+low}\\ \text{+central}\end{bmatrix} \Rightarrow \text{C:C} \_\_ \begin{bmatrix}\text{+voi}\\ \text{+cont}\\ \text{+back}\end{bmatrix} : \begin{bmatrix}\text{+voi}\\ \text{+cont}\\ \text{+back}\end{bmatrix}$$

Example: QEm@rIC → QEm6rIC, kOnf@rEnts → kOnf6rEnts

Stem-final mixed junction

The rule of r-coalescence tells us that in a sequence consisting of a long vowel plus /r/ plus /@/, the /r/ may coalesce with the vowel, yielding a falling diphthong with a lax first element. Note that this includes resyllabification since the /@/ sound, which carries syllabicity, is deleted. The result is either a monosyllable where the nasal occupies the coda, or a syllabic nasal.

(30) R-coalescence

$$\begin{bmatrix}+\text{voi}\\+\text{cont}\\+\text{back}\end{bmatrix}:0 \Rightarrow \begin{bmatrix}+\text{vowel}\\+\text{long}\\\alpha\ \text{place}\end{bmatrix}:\begin{bmatrix}+\text{vowel}\\+\text{central}\\\alpha\ \text{place}\end{bmatrix}\underline{\hspace{2em}}\begin{bmatrix}+\text{vowel}\\+\text{mid}\\+\text{central}\end{bmatrix}:\begin{bmatrix}+\text{vowel}\\+\text{low}\\+\text{central}\end{bmatrix}$$

Example: h2:r@n → h96n

Again, we need a special rule for /E/:

$$\begin{bmatrix}+\text{voi}\\+\text{cont}\\+\text{back}\end{bmatrix}:0 \Rightarrow \begin{bmatrix}+\text{vowel}\\+\text{low}\\+\text{front}\\-\text{round}\\+\text{long}\end{bmatrix}:\begin{bmatrix}+\text{vowel}\\+\text{low}\\+\text{front}\\-\text{round}\\-\text{long}\end{bmatrix}\underline{\hspace{2em}}\begin{bmatrix}+\text{vowel}\\+\text{mid}\\+\text{central}\end{bmatrix}:\begin{bmatrix}+\text{vowel}\\+\text{low}\\+\text{central}\end{bmatrix}$$

Example: vE:r@n → vE6n

Stem-final consonant junction

The following rule accounts for schwa-deletion and cluster reduction taking place in one pass. This is necessary because our declarative approach does not permit rules to apply sequentially. If this were the case, the forms described by (31) could be generated by cluster reduction applying after schwa-deletion or vice versa.

(31) Stem-final cluster simplification

$$\begin{bmatrix}-\text{voi}\\-\text{cont}\\+\text{cor}\end{bmatrix}:0 \Rightarrow \begin{bmatrix}-\text{voi}\\+\text{cont}\\+\text{cor}\end{bmatrix}:\begin{bmatrix}-\text{voi}\\+\text{cont}\\+\text{cor}\end{bmatrix}\underline{\hspace{2em}}\begin{bmatrix}+\text{vowel}\\+\text{mid}\\+\text{central}\end{bmatrix}:0$$

Example: nE:Cst@n → nE:Csn, krIst@l → krIsl

Stem-final coda

If a stem ends in /N/, the velar nasal can be reinforced, yielding [Nk]. However, the velar nasal must remain entirely within the same syllable, which requires that the suffix begin with a consonant. Suffixes beginning with a vowel, such as /UN/, would render the nasal ambisyllabic, preventing the possibility of inserting [k]: [lENUN] but not *[lENkUN] for *Längung*.

(32) Velar Reinforcement

$$0 : \begin{bmatrix} \text{-voi} \\ \text{-cont} \\ \text{+back} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{+nas} \\ \text{+back} \end{bmatrix} : \begin{bmatrix} \text{+nas} \\ \text{+back} \end{bmatrix} \text{---}$$

Example: laN → laNk

Stem-final rhyme

Stem-final vowel + *g* sequences usually have the canonical pronunciation /k/ for the final *g*. However, it can be realised as a velar or palatal fricative, depending on the place features of the preceding vowel. A front vowel entails the realisation as a palatal fricative, back vowels trigger velar realisation. It is generally assumed in the literature (e.g. Hall (1992), Walther & Wiese (1993)) that the orthographic *g* corresponds to a phonologically underlying /G/, which is fricativised and rendered voiceless by final devoicing. We have decided to include a diacritic *G* in our rule which indicates this underlying phoneme. This symbol would equally have to be present in the canonical transcription of relevant words, which could be achieved by adding a special mechanism to grapheme-to-phoneme conversion components. Again, the rule is not operative if the /G/ is shifted into the onset of a following syllable in the case of suffixisation.

(33) Velar fricativisation

$$\begin{bmatrix} \text{-voi} \\ \text{-cont} \\ \text{+back} \end{bmatrix} : \begin{bmatrix} \text{-voi} \\ \text{+cont} \\ \alpha \text{ place} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{+vowel} \\ \alpha \text{ place} \end{bmatrix} : \begin{bmatrix} \text{+vowel} \\ \alpha \text{ place} \end{bmatrix} \text{---}$$

where [α place] = {front,back}

Example: tsu:k → tsUx, b@le:k → b@le:C

However, a process which is almost the reverse, i.e. the realisation of /C/ as /k/ also takes place, viz. in words which orthographically end in *ig*. Again, words whith a different orthographical representation are not possible candidates for this change: *[drIlIk] for /drIlIC/ (*Drillich*). The index C indicates the orthographical representation *ig*.

(34) Final /k/ - /g/ alternation

$$\begin{bmatrix} \text{-voi} \\ \text{+cont} \\ \text{+front} \end{bmatrix} : \begin{bmatrix} \text{-voi} \\ \text{-cont} \\ \text{+back} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{+vowel} \\ \text{+front} \\ \text{+high} \\ \text{+central} \\ \text{-round} \end{bmatrix} : \begin{bmatrix} \text{+vowel} \\ \text{+front} \\ \text{+high} \\ \text{+central} \\ \text{-round} \end{bmatrix} \_\_$$

Example: ve:nIC → ve:nIk

Word and morph boundaries

The phenomena taking place at morph boundaries and word boundaries are nearly identical. In addition to the kind of voicing assimilation which is also to be found at syllable boundaries (i.e. a voiceless consonant becomes voiced), a different kind of voicing assimilation may take place, viz. spreading of voicelessness. This is due to the fact that voiced and voiceless consonants only come together at morph and word boundaries; remember that the German onset only allows combinations of voiceless consonants and sonorants (which do not devoice phonemically) and that voicing distinctions are neutralised in the coda. Further variant rules are cross-syllable nasal assimilation, place assimilation of /t/ and geminate simplification.

(35) Assimilation of voicelessness

$$[+\text{voi}]:[-\text{voi}] \Rightarrow [-\text{voi}]:[-\text{voi}]\ \_\_\ =:=$$

Example: opvo:l → opfo:l

(36) Cross-syllable nasal assimilation

$$\begin{bmatrix} \text{+nas} \\ \text{+cor} \end{bmatrix} : \begin{bmatrix} \text{+nas} \\ \alpha \text{ place} \end{bmatrix} \Rightarrow =:= \_\_ \ \overset{\text{C}}{[\alpha \text{ place}]} : \overset{\text{C}}{[\alpha \text{ place}]}$$

Examples: ange:b@n → aNge:b@n, anfaN@n → amfaN@n, ?aInma:l → ?aImma:l

If two adjacent consonants share the same place and manner features, they may assimilate with respect to their voicing features. This gives rise to "geminates" (Kohler (1990)), or to a single consonant. Voicing assimilation can be described by the above rule for assimilation of voicelessness, assimilation of voicedness is taken account of by Rule 24. These rules would produce geminates. Since a declarative approach does not allow intermediate representations, we need a rule that maps sequences of consonants directly to a single consonant which possibly combines features of the two.

(37) Geminate simplification

$$\text{(a)} \begin{bmatrix}\text{-voi}\\ \alpha\ \text{place}\end{bmatrix} : \begin{bmatrix}+\text{voi}\\ \alpha\ \text{place}\end{bmatrix} \Rightarrow=:= \underline{\hspace{1.5em}} \begin{bmatrix}+\text{voi}\\ \alpha\ \text{place}\end{bmatrix} : 0$$

Example: ge:t de6 → ge:de6

$$\text{(b)} \begin{bmatrix}+\text{voi}\\ \alpha\ \text{place}\end{bmatrix} : \begin{bmatrix}\text{-voi}\\ \alpha\ \text{place}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{-voi}\\ \alpha\ \text{place}\end{bmatrix} : 0 \underline{\hspace{1.5em}} =:=$$

Example: hat de6 → hate6

(38) Place assimilation of /t/

$$\begin{bmatrix}\text{-voi}\\ \text{-cont}\\ +\text{cor}\end{bmatrix} : \begin{bmatrix}\text{-voi}\\ \text{-cont}\\ \alpha\ \text{place}\end{bmatrix} \Rightarrow=:= \underline{\hspace{1.5em}} \begin{bmatrix}\text{-cont}\\ \alpha\ \text{place}\end{bmatrix} : \begin{bmatrix}\text{-cont}\\ \alpha\ \text{place}\end{bmatrix}$$

Example: ra:tkap@ → ra:kkapp@

7 Summary and Prospects

In this paper we have described the concepts and methods involved in the construction of a knowledge base for the generation and recognition of speech variants. Phonetic rules in rewrite format were established empirically on the basis of several segment-labelled speech corpora. The rules, representing linguistic-phonetic generalisation, were then converted into two-level rules FSTs. These were in turn compiled into a single pushdown transducer incorporating a transition network with subnetworks for the various domains of rule application. Finally, possible and existing applications of this knowledge base with respect to both generation and recognition of speech variants were demonstrated.

High-performance speech recognition is perfectly feasible without phonological or speech variant rules.[12] However, it is questionable whether the refinement of statistical models in order to capture highly rule-like phonetic behaviour is actually the best approach. The results may work well for limited applications such as the Wall Street Journal recognition task. More sophisticated systems which aim at recognizing very large corpora, possibly including unattested (new) words for which no training material is available, will necessarily have to incorporate some form of stored knowledge about the relation between canonical and non-canonical forms. Moreover, knowledge about speech variants is indispensible in the context of adequate automatic labelling, where recognition of non-canonical forms rather than canonical forms is required.

Phonetic-phonological rules in speech processing have been in use for twenty years (c.f. e.g. Cohen & Mercer (1975), Shoup (1980), Jekosch & Becker (1989), Giachin et al.(1991), Hoequist & Nolan (1991)), and research in this direction is still going on (current work includes Wesenick (1994), Kipp (1994)). There seems to be a general consensus about the possible benefits and disadvantages of speech variant rules: rules may enhance recognition performance in individual cases; on the other hand, they may reduce the overall recognition rate because words which where previously recognised correctly are now confused with non-canonical variants of other words. Moreover, they enlargen the search space rather drastically. The answers to these problems seem to be
(a) not to employ all available rules but to select among them, and
(b) to constrain rules in such a way that prevents overgeneration of “false” variants.

Most of the rule systems that have been constructed so far suffer from lack of linguistic generalisation, i.e. rules refer to single segments rather than to classes of segments, and they only sporadically incorporate information about prosodic context and domains of application. Linguistic generalisations, however, are necessary to recognise unattested variants which are not covered by segment-specific rules unless the rule corpus is very large. By introducing segment classes, the rules are rendered both more explicit and compact. The integration of information about prosodic context and domains of application helps to prevent overgeneration of variants.

Apart from the disadvantages mentioned above, linguistic rule components need to be established by an expert. This procedure is costly, time-comsuming an prone to errors. A combination of the advantages of the automatic genera-

[12] Most of the top-ranking ASR systems in the ARPA evaluation do indeed not make use of any speech variant rules.

tion of rules and of expert knowledge would therefore be most desirable. The possibility of automatic induction of speech variant rules should therefore be investigated. Machine learning could then be employed in order to find speech variant rules for sublanguages (dialects). This is necessary to further improve speaker-independence.

As we observed above, the use of speech variant rules in ASR systems leads to overgeneration, even if they are linguistically constrained. If used generatively, the lexicon may be expanded with forms which are possible but highly unlikely. For instance, our phonetic knowledge base includes a rule which states that unstressed vowels may be reduced to schwa. Consequently, there will be non-canonical forms where every unstressed vowel has thus been reduced. Although these forms may occur in very casual speech, the probability of their occurrence is very low. In order to limit the number of non-canonical forms, further constraints might be exploited:

First, the speaker's dialect might provide clues as to which rules should not co-occur. Features typical of North-German variants, for instance, should not combine with features of South-German dialects (see also Hoequist & Nolan (1991)). Furthermore, top-down constraints from morphology and syntax could be employed to limit the set of non-canonical hypotheses. Thus, at the onset of a stressed phrase-initial syllable, a glottal stop is more likely to be dropped than in the middle of a word or phrase. Furthermore, rules can be assigned probabilities on the basis of their frequency of occurrence in the speech corpora used to estabish the rule set. If these tendencies prove reliable, rules can be assigned probabilities. This can easily be achieved in our case by pairing transitions in the speech variant transducer with statistical values. This option is currently being applied to the system in order to be able to parametrise its output.

8 Appendix

Featural representation of phonemes

Vowels

	voice	high	mid	low	central	front	back	round	long
i:	+	+				+		+	-
i	+	+				+		-	-
I	+	+			+	+		-	
y:	+	+				+		+	+
y	+	+				+		+	-
Y	+	+			+	+		+	
e:	+		+			+		-	+
e	+		+			+		-	-
2:	+		+			+		+	+
2	+		+			+		+	-
E	+			+		+		-	-
E:	+			+		+		-	+
9	+			+		+		+	
u:	+	+					+	+	+
u	+	+					+	+	-
U	+	+			+		+	+	
o:	+		+				+	+	+
o	+		+				+	+	-
O	+		+		+		+	+	
a	+			+			+		-
a:	+			+			+		+
@	+		+		+				
6	+			+	+				

Consonants

	voi	cont	nas	lat	affr	lab	cor	palato	front	back	glott
p	-	-				+					
t	-	-					+				
k	-	-								+	
?	-	-									+
b	+	-				+					
d	+	-					+				
g	+	-								+	
f	-	+				+					
v	+	+				+					
s	-	+					+				
z	+	+					+				
S	-	+						+			
Z	+	+						+			
C	-	+							+		
j	+	+							+		
x	-	+								+	
h	-	+									+
r	+	+								+	
l	+			+							
m	+		+			+					
n	+		+				+				
N	+		+							+	
pf	-				+	+					
ts	-				+		+				
tS	-				+			+			

Output examples of the speech variant transducer

Variants for "Qa:b@nt"

Qa:b@nt	a:b@nt
Qa:mt	a:mt
Qa:bmt	a:bmt
Qab@nt	ab@nt
Qamt	amt
Qabmt	abmt

Variants for "QInfo:baU"

QInfo:baU	Info:baU
QInfobaU	InfobaU
QInfObaU	InfObaU
QInf@baU	Inf@baU
QImfo:baU	Imfo:baU
QImfobaU	ImfobaU
QImfObaU	ImfObaU
QImf@baU	Imf@baU
QImvo:baU	Imvo:baU
QImvobaU	ImvobaU
QImvObaU	ImvObaU
QImv@baU	Imv@baU
QImpfo:baU	Impfo:baU
QImpfobaU	ImpfobaU
QImpfObaU	ImpfObaU
QImpf@baU	Impf@baU
QInvo:baU	Invo:baU
QInvobaU	InvobaU
QInvObaU	InvObaU
QInv@baU	Inv@baU

9 References

Althoff, F., Carson-Berndsen, J., Drexel, G., Gibbon, D., Hübener, K., Jost, U., Kirchhoff, K., Pampel, M., Petzold, A. & V. Strom (1995) *BELLEx3+1 - Linguistische Worterkennung unter Berücksichtigung der Prosodie.* Verbmobil Technisches Dokument 22/95

Bird, S. & T.M. Ellison (1992) *One level phonology: autosegmental representations and rules as finite-state automata*, RP 51, University of Edinburgh, Centre for Cognitive Science

Bird, S. & T.M. Ellison (1994) 'One-level phonology: autosegmental representations and rules as finite automata', *Computational Linguistics 20*, 55-90

Black, A.W., Pulman, S.J., Ritchie, G.D. & G.J. Russell (1987) 'Formalisms for morphographemic description', *Proceedings of the Third Conference of the European Chapter of the Assocation for Compuational Linguistics*, 11-18

Chomsky, N. & M. Halle (1968) *The Sound Pattern of English.* New York: Harper & Row

Church, K.W. (1987) *Phonological Parsing in Speech Recognition.* Boston: Kluwer Adacemic Publishers

Carson, J. (1988) 'Unification and transduction in computational phonology' *Proceedings COLING '88*, 106-111

Carson-Berndsen, J., Gibbon, D. & K. Knäpel (1989) *Forschungsprojekt Entwicklung phonologischer Regelsysteme und Untersuchungen zur Automatisierung der Regelerstellung für Zwecke der automatischen Spracherkennung, Final Report.* Research project financed by the Deutsche Bundespost

Carson-Berndsen, J. & D. Gibbon (1992) 'Event relations at the phonetics/phonology interface', *Proceedings of the 15th International Conference on Computational Linguistics (COLING) 92*, Nantes

Carson-Berndsen, J. (1990) 'Phonological Processing of Speech Variants', *Proceedings of COLING '90*, Helsinki University, 1-4

Carson-Berndsen, J. (1992) *An event-based phonotactics for German*, ASL-TR-92/UBI

Carson-Berndsen, J. (1993) *Time Map Phonology and the Projection Problem in Spoken Language Recognition*, PhD Thesis, University of Bielefeld

Cohen, P.S. & R.L. Mercer (1975) 'The phonological component of an automatic speech-recognition system' in Reddy, R. (ed.) *Speech Recognition*, New York: Academic Press

Gibbon, D, (1987) 'Finite state processing of tone systems', *Proceedings of the Third European conference of the Association for Computational Linguistics*, 291-297

Gibbon, D., Bleiching, D, Carson-Berndsen, J., Langer, H., & M. Pampel (1992) *BELLEx3 - Bielefeld Engine for Lattice-to-Lattice Event- Parsing*, Technical Report, University of Bielefeld

Giachin, E.P., Rosenberg, A.E. & C. Lee (1991) 'Word juncture modeling using phonological rules for HMM-based continuous speech recognition', *Computer, Speech and Language 5*, 155-168

Hall, T.A. (1992) *Syllable Structure and Syllable-Related Processes in German.* Tübingen: Niemeyer

Hübener, K. (1993) *Detektion akustisch-phonetischer Ereignisse.* Verbmobil-Memo 5/93, University of Hamburg, Department of Computer Science

Hübener, K. & J. Carson-Berndsen (1994) *Phoneme Recognition using Acoustic Events*, Vermbobil Technical Report No. 15

Hoequist, C. & F. Nolan (1991) 'On an application of phonological knowledge in automatic speech recognition', *Computer, Speech and Language 5*, 133-153

Hopcroft, J.E. & J.D. Ullman (1979) *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley

Jekosch, U. & T. Becker (1989) 'Maschinelle Generierung von Aussprachevarianten: Perspektiven für Sprachsynthese- und Spracherkennungssysteme' *Informationstechnik* **it** *31* , 400-406

Johnson, C.D (1972) *Formal Aspects of Phonological Description.* The Hague: Mouton

Kaplan, R.M. & M. Kay (1981) 'Phonological rules and finite-state transducers', paper presented to the *Winter meeting of the Linguistic Society of America*, New York

Kaplan, R.M. & M. Kay (1994) 'Regular models of phonological rule systems', *Computational Linguistics 20*, 331-378

Kartunnen, L. (1983) 'KIMMO: a general morphological processor', *Texas Linguistic Forum 22*, 165-186

Kartunnen, L. & K. Wittenburg (1983) 'A two-level morphological analysis of English', *Texas Linguistic Forum 22*, 217-228

Kay, M. (1983) 'When meta-rules are not meta-rules' in Sparck Jones, K. & Y. Wilks (eds.) *Automatic Natural Language Parsing*, Chichester/New York: Ellis Horwood/Wiley

Kay, M. (1987) 'Nonconcatenative finite-state morphology', *Proceedings of the Third European Conference of the Association for Computational Linguistics*, 2-10

Kipp, A. (1994) *Automatische Generierung von Aussprachevarianten und deren Anwendung in der Spracherkennung*, Diploma Thesis, TU München

Kohler, K. (1974) 'Koartikulation und Steuerung im Deutschen' in *Sprachsystem und Sprachgebrauch. Festschrift für H. Moser, Teil 1*, Düsseldorf: Schwann, 172-192

Kohler, K. (1979) 'Kommunikative Aspekte satzphonetischer Prozesse' in H. Vater (ed.) *Phonologisch Probleme des Deuschen, Studien zur deutschen Grammatik 10*, Tübingen: Narr, 13-39

Kohler, K. (1990) 'Segmental reduction in connected speech in German: phonological facts and phonetic explanations' in Hardcastle, W.J. & A. Marchal (eds.) *Speech Production and Speech Modelling*, Dordrecht: Kluwer, 69-92

Kohler, K. (1994) 'Glottal stops and glottalization in German', *Phonetica 51*, 38-51

Koskenniemi, K. (1983) *Two-level morphology.* PhD Thesis, University of Helsinki

Kröger, B.J. (1993) 'A gestural production model and its application to reduction in German', *Phonetica 50*, 213-233

Pulman, S.G. & M.R. Hepple (1993) 'A feature-based formalism for two-level phonology: a description and implementation', *Computer, Speech and Language 7*, 333-358

Ritchie, G.D., Black, A.W., Pulman, S.J. & G.J. Russell (1987) *The Edinburgh-Cambridge Morphological Analyser and Dictionary System. (Prototype: Version 2.4) User Manual*, Cambridge University Computer Laboratory

Roche, E. & Y. Schabes (1994) *Deterministic Part-Of-Speech Tagging With Finite State Transducers*, Technical Report 94-07, Mitsubishi Electric Research Laboratories, Cambridge, MA

Shoup, J.E. (1980) 'Phonological aspects of speech recognition' in Lea, W. (ed.) *Trends in Speech Recognition*, New Jersey: Prentice Hall, 125-138

Walther, M. & R. Wiese (1993) 'Deklarative vs. prozedurale Modellierung von Konsonantenalternationen im Deutschen', *Proceedings of DGfS/ CL '93*, University of Hamburg

Wells, J.C. (1989) 'Computer-coded phonemic notation of individual language of the European Community', *Journal of the International Phonetic Association 19(1)*, 31-54

Wesenick, B. (1994) *Entwurf eines überspezifizierenden Regelsystems der Aussprache des Deutschen als Basis für empirische Untersuchungen*, MA thesis, Universität München