

# Name pronunciation in German text-to-speech synthesis

**Stefanie Jannedy**

Linguistics Dept.  
Ohio State University  
Columbus, OH 43210, USA  
jannedy@ling.ohio-state.edu

**Bernd Möbius**

Language Modeling Research  
Bell Laboratories  
Murray Hill, NJ 07974, USA  
bmo@research.bell-labs.com

## Abstract

We describe the name analysis and pronunciation component in the German version of the Bell Labs multilingual text-to-speech system. We concentrate on street names because they encompass interesting aspects of geographical and personal names. The system was implemented in the framework of finite-state transducer technology, using linguistic criteria as well as frequency distributions derived from a database. In evaluation experiments, we compared the performances of the general-purpose text analysis and the name-specific system on training and test materials. The name-specific system significantly outperforms the generic system. The error rates compare favorably with results reported in the research literature. Finally, we discuss areas for future work.

## 1 Introduction

The correct pronunciation of names is one of the biggest challenges for text-to-speech (TTS) conversion systems. At the same time, many current or envisioned applications, such as reverse directory systems, automated operator services, catalog ordering or navigation systems, to name just a few, crucially depend upon an accurate and intelligible pronunciation of names. Besides these specific applications, any kind of well-formed text input to a general-purpose TTS system is extremely likely to contain names, and the system has to be well equipped to process these names. This requirement was the main motivation to develop a name analysis and pronunciation component for the German version of the Bell Labs multilingual text-to-speech system (GerTTS) (Möbius et al., 1996).

Names are conventionally categorized into personal names (first and surnames), geographical names (place, city and street names), and brand names (organization, company and product names).

In this paper, we concentrate on street names because they encompass interesting aspects of geographical as well as of personal names. Linguistic descriptions and criteria as well as statistical considerations, in the sense of frequency distributions derived from a large database, were used in the construction of the name analysis component. The system was implemented in the framework of finite-state transducer (FST) technology (see (Sproat, 1992) for a discussion focussing on morphology). For evaluation purposes, we compared the performances of the general-purpose text analysis and the name-specific systems on training and test materials.

As of now, we have neither attempted to determine the etymological or ethnic origin of names, nor have we addressed the problem of detecting names in arbitrary text. However, due to the integration of the name component into the general text analysis system of GerTTS, the latter problem has a reasonable solution.

## 2 Some problems in name analysis

What makes name pronunciation difficult, or special, in comparison to words that are considered as regular entries in the lexicon of a given language? Various reasons are given in the research literature (Carlson, Granström, and Lindström, 1989; Macchi and Spiegel, 1990; Vitale, 1991; van Coile, Leys, and Mortier, 1992; Coker, Church, and Liberman, 1990; Belhoula, 1993):

- Names can be of very diverse etymological origin and can surface in another language without undergoing the slow linguistic process of assimilation to the phonological system of the new language.
- The number of distinct names tends to be very large: For English, a typical unabridged collegiate dictionary lists about 250,000 word types, whereas a list of surnames compiled from an address database contains 1.5 million types (72 million tokens) (Coker, Church, and Liberman, 1990). It is reasonable to assume similar ratios

for German, although no precise numbers are currently available.

- There is no exhaustive list of names; and in German and some related Germanic languages, street names in particular are usually constructed like compounds (*Rheinstraße*, *Kennedyallee*) which makes decomposition both practical and necessary.
- Name pronunciation is known to be idiosyncratic; there are many pronunciations contradicting common phonological patterns, as well as alternative pronunciations for certain grapheme strings.
- In many languages, general-purpose grapheme-to-phoneme rules are to a significant extent inappropriate for names (Macchi and Spiegel, 1990; Vitale, 1991).
- Names are not equally amenable to morphological processes, such as word formation and derivation or to morphological decomposition, as regular words are. That does not render such an approach unfeasible, though, as we show in this paper.
- The large number of different names together with a restricted morphological structure leads to a coverage problem: It is known that a relatively small number of high-frequency words can cover a high percentage of word tokens in arbitrary text; the ratio is far less favorable for names (Carlson, Granström, and Lindström, 1989; van Coile, Leys, and Mortier, 1992).

We will now illustrate some of the idiosyncracies and peculiarities of names that the analysis has to cope with. Let us first consider morphological issues. Some German street names can be morphologically and lexically analyzed, such as *Kurfürst-en-damm* ('electorial prince dam'), *Kirche-n-weg* ('church path'). Many, however, are not decomposable, such as *Heumerich* ('?') or *Rimparstraße* ('?Rimpar street'), at least not beyond obvious and unproblematic components (*Straße*, *Weg*, *Platz*, etc.).

Even more serious problems arise on the phonological level. As indicated above, general-purpose pronunciation rules often do not apply to names. For instance, the grapheme <e> in an open stressed syllable is usually pronounced [e:]; however, in many first names (*Stefan*, *Melanie*) it is pronounced [ɛ]. Or consider the word-final grapheme string <ie> in *Batterie* [batə'ri:] 'battery', *Materie* [mat'e:riə] 'matter', and the name *Rosemarie* [r'o:zəma:ri:]. And word-final <us>: *Mus* [m'u:s] 'mush, jam' vs. *Erasmus* [er'asmus]. A more special and yet typical example: In regular German words the morpheme-initial substring <chem> as in *chemisch* is pronounced [çe:m], whereas in the name of the city *Chemnitz* it is pronounced [kem].

Generally speaking, nothing ensures correct pronunciation better than a direct hit in a pronunciation dictionary. However, for the reasons detailed above this approach is not feasible for names. In short, we are not dealing with a memory or storage problem but with the requirement to be able to approximately correctly analyze unseen orthographic strings. We therefore decided to use a weighted finite-state transducer machinery, which is the technological framework for the text analysis components of the Bell Labs multilingual TTS system. FST technology enables the dynamic combination and recombination of lexical and morphological substrings, which cannot be achieved by a static pronunciation dictionary. We will now describe the procedure of collecting lexically or morphologically meaningful graphemic substrings that are used productively in name formation.

### 3 Productive name components

#### 3.1 Database

Our training material is based on publically available data extracted from a phone and address directory of Germany. The database is provided on CD-ROM (D-Info, 1995). It lists all customers of Deutsche Telekom by name, street address, city, phone number, and postal code. The CD-ROM contains data retrieval and export software. The database is somewhat inconsistent in that information for some fields is occasionally missing, more than one person is listed in the name field, business information is added to the name field, first names and street names are abbreviated. Yet, due to its listing of more than 30 million customer records it provides an exhaustive coverage of name-related phenomena in German.

#### 3.2 City names

The data retrieval software did not provide a way to export a complete list of cities, towns, and villages; thus we searched for all records listing city halls, township and municipality administrations and the like, and then exported the pertinent city names. This method yielded 3,837 city names, approximately 15% of all the cities (including urban districts) covered in the database. It is reasonable to assume, however, that this corpus provided sufficient coverage of lexical and morphological subcomponents of city names.

We extracted graphemic substrings of different lengths from all city names. The length of the strings varied from 3 to 7 graphemes. Useful substrings were selected using frequency analysis (automatically) and native speaker intuition (manually). The final list of morphologically meaningful substrings consisted of 295 entries. In a recall test, these 295 strings accounted for 2,969 of the original list of city names, yielding a coverage of  $2,969/3,837 = 77.4\%$ .

|                  | München<br>(south) | Berlin<br>(east) | Hamburg<br>(north) | Köln<br>(west) | Total  |
|------------------|--------------------|------------------|--------------------|----------------|--------|
| component types  | 7,127              | 7,291            | 8,027              | 4,396          | 26,841 |
| morphemes        | 922                | 574              | 320                | 124            | 1,940  |
| recall           | 2,387              | 2,538            | 4,214              | 2,102          | 11,241 |
| residuals (abs.) | 4,740              | 4,753            | 3,813              | 2,294          | 15,600 |
| residuals (rel.) | 66.5%              | 65.0%            | 47.5%              | 52.2%          | 58.1%  |

Table 1: Extraction of productive street name components: quantitative data.

### 3.3 First names

The training corpus for first names and street names was assembled based on data from the four largest cities in Germany: Berlin, Hamburg, Köln (Cologne) and München (Munich). These four cities also provide an approximately representative geographical and regional/dialectal coverage. The size and geography criteria were also applied to the selection of the test material which was extracted from the cities of Frankfurt am Main and Dresden (see Evaluation).

We retrieved all available first names from the records of the four cities and collected those whose frequency exceeded 100. To this corpus we added the most popular male and female (10 each) names given to newborn children in the years 1995/96, in both the former East and West Germany, according to an official statistical source on the internet. The corpus also contains interesting spelling variants (*Helmut/Hellmuth*) as well as peculiarities attributable to regional tastes and fashions (*Maik, Maia*). The total number of first names in our list is 754.

No attempt was made to arrive at some form of morphological decomposition despite several obvious recurring components, such as <-hild>, <-bert>, <-fried>; the number of these components is very small, and they are not productive in name-forming processes anymore.

### 3.4 Streets

We retrieved all available street names from the records of the four cities. The street names were split up into their individual word-like components, i.e., a street name like *Konrad-Adenauer-Platz* created three separate entries: *Konrad*, *Adenauer*, and *Platz*. This list was then sorted and made unique.

The type inventory of street name components was then used to collect lexically and semantically meaningful components, which we will henceforth conveniently call ‘morphemes’. In analogy to the procedure for city names, these morphemes were used in a recall test on the original street name component type list. This approach was successively applied to the street name inventory of the four cities, starting with München, exploiting the result of this first round in the second city, Berlin, applying the combined result of this second round on the third

city, and so on.

Table 1 gives the numbers corresponding to the steps of the procedure just described. The number of morphemes collected from the four cities is 1,940. The selection criterion was frequency: Component types occurring repeatedly within a city database were considered as productive or marginally productive. The 1,940 morphemes recall 11,241 component types out of the total of 26,841 (or 41.9%), leaving 15,600 types (or 58.1%) that are unaccounted for (‘residuals’) by the morphemes.

Residuals that occur in at least two out of four cities (2,008) were then added to the list of 1,940 morphemes. The reasoning behind this is that there are component types that occur exactly once in a given city but do occur in virtually every city. To give a concrete example: There is usually only one *Hauptstraße* (‘main street’) in any given city but you almost certainly do find a *Hauptstraße* in every city. After some editing and data clean-up, the final list of linguistically motivated street name morphemes contained 3,124 entries.

## 4 Compositional model of street names

In this section we will present a compositional model of street names that is based on a morphological word model and also includes a phonetic syllable model. We will also describe the implementation of these models in the form of a finite-state transducer.

### 4.1 Naming schemes for streets in German

Evidently, there is a finite list of lexical items that almost unambiguously mark a name as a street name; among these items are *Straße*, *Weg*, *Platz*, *Gasse*, *Allée*, *Markt* and probably a dozen more. These *street name markers* are used to construct street names involving persons (*Stephan-Lochner-Straße*, *Kennedyallee*), geographical places (*Tübinger Allee*), or objects (*Chrysanthemenweg*, *Containerbahnhof*); street names with local, regional or dialectal peculiarities (*Söbendieken*, *Höglstieg*); and finally intransparent street names (*Krüsistraße*, *Damaschkestraße*). Some names of the latter type may actually refer to persons’ names but the origin is not transparent to the native speaker.

```

START ROOT {Eps}
ROOT FIRST SyllModel

.
.
.
ROOT FIRST 'Alfons{firstname}<0.2>
ROOT FIRST D'irk{firstname}<0.2>
ROOT FIRST D'ominik<{firstname}0.2>

.
.
.
ROOT FIRST b'urg{city}
ROOT FIRST br'uck{city}<0.2>
ROOT FIRST d'orf{city}
ROOT FIRST fl'eet{city}<0.2>

.
.
.
ROOT FIRST d'ach{street}<0.2>
ROOT FIRST h'ecke{street}<0.2>
ROOT FIRST kl'ar{street}<0.2>
ROOT FIRST kl'ee{street}<0.2>
ROOT FIRST kl'ein{street}<0.2>
ROOT FIRST st'ein{street}<0.2>

.
.
.
ROOT FIRST all'ee{marker}
ROOT FIRST g'arten{marker}
ROOT FIRST pl'atz{marker}
ROOT FIRST w'eg{marker}

.
.
.
FIRST ROOT {++}<0.1>
FIRST FUGE {Eps}<0.2>
FIRST FUGE s<0.2>
FIRST FUGE n<0.2>
FIRST SUFFIX {Eps}<0.2>
FIRST SUFFIX s<0.2>
FIRST SUFFIX n<0.2>
FUGE FIRST {Eps}<10.0>
FUGE ROOT {++}<0.5>
SUFFIX END {name}
END

```

Figure 1: Parts of a grammar (in *arclist* format) for street name decomposition in German.

#### 4.2 Building a generative transducer for street names

The component types collected from the city, first name and street databases were integrated into a combined list of 4,173 productive name components: 295 from city names, 754 from first names, 3,124 from street names. Together with the basic street name markers, these components were used to construct a name analysis module. The module was implemented as a finite-state transducer using Richard Sproat’s *lextools* (Sproat, 1995), a toolkit for creating finite-state machines from linguistic descriptions. The module is therefore compatible with the other text analysis components in the German TTS system (Möbius, 1997) that were all developed in the same FSM technology framework.

One of the *lextools*, the program *arclist*, is particularly well suited for name analysis. The tool facilitates writing a finite-state grammar that describes words of arbitrary morphological complexity and length (Sproat, 1995). In the TTS system it is

also applied to the morphological analysis of compounds and unknown words.

Figure 1 shows parts of the *arclist* source file for street name decomposition. The arc which describes the transition from the initial state “START” to the state “ROOT” is labeled with  $\varepsilon$  (Epsilon, the empty string). The transition from “ROOT” to the state “FIRST” is defined by three large families of arcs which represent the lists of first names, productive city name components, and productive street name components, respectively, as described in the previous section.

The transition from “ROOT” to “FIRST” which is labeled *SyllModel* is a place holder for a phonetic syllable model. This syllable model reflects the phonotactics and the segmental structure of syllables in German, or rather their correlates on the orthographic surface. This allows the module to analyze substrings of names that are unaccounted for by the explicitly listed name components (see ‘residuals’ in the previous section) in arbitrary locations in a complex name. A detailed discussion of the syllable model is presented elsewhere (Möbius, 1997).

From the state “FIRST” there is a transition back to “ROOT”, either directly or via the state “FUGE”, thereby allowing arbitrarily long concatenations of name components. Labels on the arcs to “FUGE” represent infixes (‘Fugen’) that German word forming grammar requires as insertions between components within a compounded word in certain cases, such as *Wilhelm+s+platz* or *Linde+n+hof*. The final state “END” can only be reached from “FIRST” by way of “SUFFIX”. This transition is defined by a family of arcs which represents common inflectional and derivational suffixes. On termination the word is tagged with the label ‘name’ which can be used as part-of-speech information by other components of the TTS system.

Most arc labels are weighted by being assigned a cost. Weights are a convenient way to describe and predict linguistic alternations. In general, such a description can be based on an expert’s analysis of linguistic data and his or her intuition, or on statistical probabilities derived from annotated corpora. Works by Riley (Riley, 1994) and Yarowsky (Yarowsky, 1994) are examples of inferring models of linguistic alternation from large corpora. However, these methods require a database that is annotated for all relevant factors, and levels on these factors. Despite our large *raw* corpus, we lack the type of database resources required by these methods. Thus, all weights in the text analysis components of GerTTS are currently based on linguistic intuition; they are assigned such that after integration of the name component in the general text analysis system, direct hits in the general-purpose lexicon will be less expensive than name analyses (see Discussion). No weights or costs are assigned to the most frequently occurring street name components, previously intro-

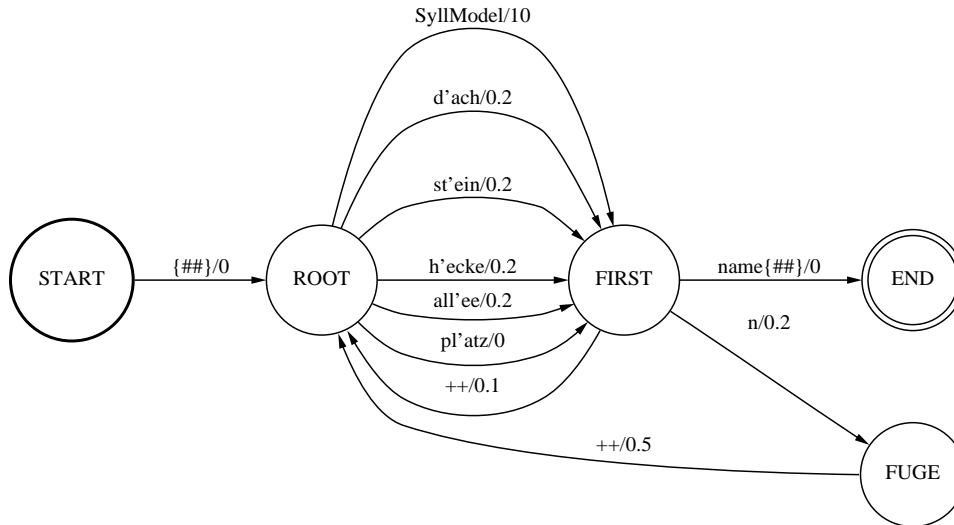


Figure 2: The transducer compiled from the sub-grammar that performs the decomposition of the fictitious street name *Dachsteinhohenheckenalleenplatz*.

duced as street name markers, making them more likely to be used during name decomposition. The orthographic strings are annotated with symbols for primary (') and secondary (") lexical stress. The symbol {++} indicates a morpheme boundary.

The finite-state transducer that this grammar is compiled into is far too complex to be usefully diagrammed here. For the sake of exemplification, let us instead consider the complex fictitious street name *Dachsteinhohenheckenalleenplatz*. Figure 2 shows the transducer corresponding to the sub-grammar that performs the decomposition of this name. The path through the graph is as follows:

The arc between the initial state “START” and “ROOT” is labeled with a word boundary {##} and zero cost (0). From here we take the arc with the label *d'ach* and a cost of 0.2 to state “FIRST”. The next name component that can be found in the grammar is *stein*; we have to return to “ROOT” by way of an arc that is labeled with a morph boundary and a cost of 0.1. The next known component is *hecke*, leaving a residual string *hohen* which has to be analyzed by means of the syllable model. Applying the syllable model is expensive because we want to cover the name string with as many known components as possible. The costs actually vary depending upon the number of syllables in the residual string and the number of graphemes in each syllable; the string *hohen* would thus have to be decomposed into a root *hohe* and the ‘Fuge’ *n*. For the sake of simplicity we assign a flat cost of 10.0 in our toy example. In the transition between *hecke* and *alleen* a ‘Fuge’ (*n*) has to be inserted. The cost of the following morph boundary is higher (0.5) than usual in order to favor components that do not require infixation. Another *Fuge* has to be inserted after *alleen*. The cost

of the last component, *platz*, is zero because this is one of the customary street name markers. Finally, the completely analyzed word is tagged as a *name*, and a word boundary is appended on the way to the final state “END”.

The morphological information provided by the name analysis component is exploited by the phonological or pronunciation rules. This component of the linguistic analysis is implemented using a modified version of the Kaplan and Kay rewrite rule algorithm (Kaplan and Kay, 1994).

## 5 Evaluation

We evaluated the name analysis system by comparing the pronunciation performance of two versions of the TTS system, one with and one without the name-specific module. We ran both versions on two lists of street names, one selected from the training material and the other from unseen data.

### 5.1 General-purpose vs. name-specific analysis

Two versions of the German TTS system were involved in the evaluation experiments, differing in the structure of the text analysis component. The first system contained the regular text analysis modules, including a general-purpose module that handles words that are not represented in the system’s lexicon: typically compounds and names. This version will be referred to as the *old* system. The second version purely consisted of the name grammar transducer discussed in the previous section. It did not have any other lexical information at its disposal. This version will be referred to as the *new* system.

|   | Training Data   | Test Data      |
|---|-----------------|----------------|
| number of names                           | 631             | 206            |
| at least one system wrong                 | 250/631 (39.6%) | 82/206 (39.8%) |
| both systems wrong                        | 72/250 (28.8%)  | 26/ 82 (31.7%) |
| total error rate<br>(no correct solution) | 72/631 (11.1%)  | 26/206 (12.6%) |

Table 2: Performance of the general-purpose and the name-specific text analysis systems on training and test data sets.

|   | Training Data   | Test Data     |
|---|-----------------|---------------|
| new system correct<br>&& old system wrong | 138/163 (84.7%) | 35/50 (70.0%) |
| old system correct<br>&& new system wrong | 25/163 (15.3%)  | 15/50 (30.0%) |
| net improvement                           | 113/163 (69.4%) | 20/50 (40.0%) |

Table 3: Comparison between the general-purpose and the name-specific text analysis systems on training and test data sets.

## 5.2 Training vs. test materials

The textual materials used in the evaluation experiments consisted of two sets of data. The first set, henceforth *training data*, was a subset of the data that were used in building the name analysis grammar. For this set, the street names for each of the four cities Berlin, Hamburg, Köln and München were randomized. We then selected every 50th entry from the four files, yielding a total of 631 street names; thus, the training set also reflected the respective size of the cities.

The second set, henceforth *test data*, was extracted from the databases of the cities Frankfurt am Main and Dresden. Using the procedure described above, we selected 206 street names. Besides being among the ten largest German cities, Frankfurt and Dresden also meet the requirement of a balanced geographical and dialectal coverage. These data were not used in building the name analysis system.

## 5.3 Results

The old and the new versions of the TTS system were run on the training and the test set. Pronunciation performance was evaluated on the symbolic level by manually checking the correctness of the resulting transcriptions. A transcription was considered correct when no segmental errors or erroneous syllabic stress assignments were detected. Multiple mistakes within the same name were considered as one error. Thus, we made a binary decision between correct and incorrect transcriptions.

Table 2 summarizes the results. On the training data, in 250 out of a total of 631 names (39.6%) at least one of the two systems was incorrect. In 72

out of these 250 cases (28.8%) both systems were wrong. Thus, for 72 out of 631 names (11.4%) no correct transcription was obtained by either system.

On the test data, at least one of the two systems was incorrect in 82 out of a total of 206 names (39.8%), an almost identical result as for the training data. However, in 26 out of these 82 cases (31.7%) both systems were wrong. In other words, no correct transcription was obtained by either system for 26 out of 206 names (12.6%), which is only slightly higher than for the training data.

Table 3 compares the performances of the two text analysis systems. On the training data, the new system outperforms the old one in 138 of the 163 cases (84.7%) where one of the systems was correct and the other one was wrong; we disregard here all cases where both systems were correct as well as the 87 names for which no correct transcription was given by either system. But there were also 25 cases (15.3%) where the old system outperformed the new one. Thus, the net improvement by the name-specific system over the old one is 69.4%.

On the test data set, the old system gives the correct solution in 15 of 50 cases (30.0%), compared to 35 names (70.0%) for which the new system gives the correct transcription; again, all cases were excluded in which both systems performed equally well or poorly. The net improvement by the name-specific system over the generic one on the test data is thus 40%.

A detailed error analysis yielded the following types of remaining problems:

- Syllabic stress: *Saarbrücken* [za:əbR'ʏkən] but *Zweibrücken* [tsv'aɪbRʏkən].

- Vowel quality: *Soest* [zo:st], not [zø:st] or [zo:əst].
- Consonant quality: *Chemnitz* [kɛmɲits], not [çɛ:mnits] in analogy to *chemisch* [çɛ:miʃ].
- Morphology: Erroneous decomposition of substrings (hyper-correction over old system); e.g., *Rim+par+straße* [ri:mpæ] instead of *Rim-par+straße* [ri:mpæ].
- Pronunciation rules: “Holes” in the general-purpose pronunciation rule set were revealed by orthographic substrings that do not occur in the regular lexicon. It has been shown for English (van Santen, 1992) that the frequency distribution of triphones in names is quite dissimilar to the one found in regular words.
- Idiosyncrasies: Peculiar pronunciations that cannot be described by rules and that even native speakers quite often do not know or do not agree upon; e.g., *Oeynhausēn* [ø:nhauzən], *Duisdorf* [dy:sdœf] or [du:sdœf] or [du:isdœf].

## 6 Discussion and future work

After the evaluation, the name analysis transducer was integrated into the text analysis component of the German TTS system. The weights were adjusted in such a way that for any token, i.e., word or word form, in the input text an immediate match in the lexicon is always favored over name analysis which in turn is preferred to unknown word analysis. Even though the evaluation experiments reported in this paper were performed on names in isolation rather than in sentential contexts, the error rates obtained in these experiments (Table 2) correspond to the performance on names by the integrated text analysis component for arbitrary text.

There are two ways of interpreting the results. On the one hand, despite a significant improvement over the previous general-purpose text analysis we have to expect a pronunciation error rate of 11–13% for unknown names. In other words, roughly one out of eight names will be pronounced incorrectly.

On the other hand, this performance compares rather favorably with the results reported for the German branch of the European Onomastica project (Onomastica, 1995). Onomastica was funded by the European Community from 1993 to 1995 and aimed to produce pronunciation dictionaries of proper names and place names in eleven languages. The final report describes the performance of grapheme-to-phoneme rule sets developed for each language. For German, the accuracy rate for quality band III—names which were transcribed by rule only—was 71%; in other words, the error rate in the same sense as used in this paper was 29%. The grapheme-to-phoneme conversion rules were written by experts, based on tens of thousands of the most frequent

names that were manually transcribed by an expert phonetician.

However, the Onomastica results can only serve as a qualitative point of reference and should not be compared to our results in a strictly quantitative sense, for the following reasons. First, the percentage of proper names is likely to be much higher in the Onomastica database (no numbers are given in the report), in which case higher error rates should be expected due to the inherent difficulty of proper name pronunciation. In our study, proper names were only covered in the context of street names. Second, Onomastica did not apply morphological analysis to names, while morphological decomposition, and word and syllable models, are the core of our approach. Third, Onomastica developed name-specific grapheme-to-phoneme rule sets, whereas we did not augment the general-purpose pronunciation rules.

How can the remaining problems be solved, and what are the topics for future work? For the task of grapheme-to-phoneme conversion, several approaches have been proposed as alternatives to explicit rule systems, particularly self-learning methods (van Coile, 1990; Torkkola, 1993; Andersen and Dalsgaard, 1994) and neural networks (Sejnowski and Rosenberg, 1987; An et al., 1988). None of these methods were explored and applied in the present study. One reason is that it is difficult to construct or select a database if the set of factors that influence name pronunciation is at least partially unknown. In addition, even for an initially incomplete factor set the corresponding feature space is likely to cause coverage problems; neural nets, for instance, are known to perform rather poorly at predicting unseen feature vectors. However, with the results of the error analysis as a starting point, we feel that a definition of the factor set is now more feasible.

One obvious area for improvement is to add a name-specific set of pronunciation rules to the general-purpose one. Using this approach, Belhoula (Belhoula, 1993) reports error rates of 4.3% for German place names and 10% for last names. These results are obtained in recall tests on a manually transcribed training corpus; it remains unclear, however, whether the error rates are reported by letter or by word.

The addition of name-specific rules presupposes that the system knows which orthographic strings are names and which are regular words. The problem of name detection in arbitrary text (see (Thielen, 1995) for an approach to German name tagging) has not been addressed in our study; instead, it was by-passed for the time being by integrating the name component into the general text analysis system and by adjusting the weights appropriately.

Other areas for future work are the systematic treatment of proper names outside the context of street names, and of brand names, trademarks, and

company names. One important consideration here is the recognition of the ethnic origin of a name and the application of appropriate specific pronunciation rules. Heuristics, such as name pronunciation by analogy and rhyming (Coker, Church, and Liberman, 1990) and methods for, e.g., syllabic stress assignment (Church, 1986) can serve as role models for this ambitious task.

## 7 Acknowledgments

We wish to acknowledge Richard Sproat who developed and provided the *lextools* toolkit; this work also benefited from his advice. We also wish to thank an anonymous reviewer for constructive suggestions.

## References

- Z. An, S. Mniszewski, Y. Lee, G. Papcun, and G. Doolen. 1988. Hiertalker: A default hierarchy of high order neural networks that learns to read English aloud. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 2, pages 221–228, San Diego, CA.
- Ove Andersen and Paul Dalsgaard. 1994. A self-learning approach to transcription of Danish proper names. In *Proceedings of the International Conference on Spoken Language Processing, ICSLP-94*, volume 3, pages 1627–1630, Yokohama, Japan.
- Karim Belhoula. 1993. A concept for the synthesis of names. In *ESCA Workshop on Applications of Speech Technology*, Lautrach, Germany.
- Rolf Carlson, Björn Granström, and Anders Lindström. 1989. Predicting name pronunciation for a reverse directory service. In *Proceedings of the European Conference on Speech Communication and Technology, Eurospeech-89*, volume 1, pages 113–116, Paris, France.
- Kenneth Church. 1986. Stress assignment in letter to sound rules for speech synthesis. In *Proceedings of the IEEE International Conference on Acoustics and Speech Signal Processing, ICASSP-86*, volume 4, pages 2423–2426, Tokyo, Japan.
- Cecil H. Coker. 1990. Morphology and rhyming: Two powerful alternatives to letter-to-sound rules for speech synthesis. In *Proceedings of the ESCA Workshop on Speech Synthesis*, pages 83–86, Autrans, France.
- D-Info. 1995. D-Info—Adress- und Telefonauskunft Deutschland. CD-ROM. TopWare, Mannheim, Germany.
- Ronald Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20:331–378.
- Marian Macchi and Murray Spiegel. 1990. Using a demisyllable inventory to synthesize names. *Speech Technology*, pages 208–212.
- Bernd Möbius. 1997. Text analysis in the Bell Labs German TTS system. Technical report, Bell Laboratories.
- Bernd Möbius, Juergen Schroeter, Jan van Santen, Richard Sproat, and Joseph Olive. 1996. Recent advances in multilingual text-to-speech synthesis. In *Fortschritte der Akustik—DAGA '96*, Bad Honnef, Germany. DPG.
- Onomastica. 1995. Multi-language pronunciation dictionary of proper names and place names. Technical report, European Community, Ling. Res. and Engin. Prog. Project No. LRE-61004, Final Report, 30 May 1995.
- Michael Riley. 1994. Tree-based models of speech and language. In *Proceedings of the IEEE-IMS Workshop on Information Theory and Statistics*, Alexandria, VA.
- T. Sejnowski and C.R. Rosenberg. 1987. Parallel networks that learn to pronounce English text. *Complex Systems*, 1:144–168.
- Richard Sproat. 1992. *Morphology and computation*. MIT Press, Cambridge, MA.
- Richard Sproat. 1995. Lextools: Tools for finite-state linguistic analysis. Technical report, AT&T Bell Laboratories.
- Christine Thielen. 1995. An approach to proper name tagging in German. In *Proceedings of the EACL-95 SIGDAT Workshop: From Text to Tags*, Dublin, Ireland.
- Kari Torkkola. 1993. An efficient way to learn English grapheme-to-phoneme rules automatically. In *Proceedings of the IEEE International Conference on Acoustics and Speech Signal Processing, ICASSP-93*, volume 2, pages 199–202.
- Bert van Coile. 1990. Inductive learning of grapheme-to-phoneme rules. In *Proceedings of the International Conference on Spoken Language Processing, ICSLP-90*, volume 2, pages 765–768, Kobe, Japan.
- Bert van Coile, Steven Leys, and Luc Mortier. 1992. On the development of a name pronunciation system. In *Proceedings of the International Conference on Spoken Language Processing, ICSLP-92*, volume 1, pages 487–490, Banff, Alberta.
- Jan van Santen. 1992. Personal communication.
- Tony Vitale. 1991. An algorithm for high accuracy name pronunciation by parametric speech synthesizer. *Computational Linguistics*, 17:257–276.
- David Yarowsky. 1994. Homograph disambiguation in speech synthesis. In *Proceedings of the Second ESCA Workshop on Speech Synthesis*, pages 244–247, New Paltz, NY.