# Probabilistic Context-Free Grammars for Syllabification and Grapheme-to-Phoneme Conversion

**Karin Müller**
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
karin.mueller@ims.uni-stuttgart.de

## Abstract

We investigated the applicability of probabilistic context-free grammars to syllabification and grapheme-to-phoneme conversion. The results show that the standard probability model of context-free grammars performs very well in predicting syllable boundaries. However, our results indicate that the standard probability model does not solve grapheme-to-phoneme conversion sufficiently although, we varied all free parameters of the probabilistic reestimation procedure.

## 1 Introduction

In this paper we present an approach to unsupervised learning and automatic detection of syllable boundaries as well as automatic grapheme-to-phoneme conversion using probabilistic context-free grammars (PCFGs). A text-to-speech system (TTS), like those described in Sproat (1998), needs a module where the words are converted from graphemes to phonemes, i.e. its transcription, and one that syllabifies the obtained phoneme string before they can be further processed to speech. The two tasks can be solved both with rule-based and with probabilistic methods. Rule-based methods are facing the problem that they have to return one analysis. If there are several possible analyses then the rule-based system has the problem of disambiguation. Probabilistic methods, however, yield the most probable analysis according to the training corpus. Our approach builds on two resources. The first resource are manually constructed context-free grammars (CFGs) for both syllabification and grapheme-to-phoneme conversion (G2P). The CFG generates for the G2P task all sequences of phonemes corresponding to a given orthographic input word. For the syllabification task, the CFG generates all possible syllable boundaries. We use context-free grammars for generating transcriptions, and syllabified phoneme strings, because grammars are expressive and writing grammar-rules is easy and intuitive. We trained the CFGs on a training corpus that was extracted from a large newspaper corpus. The second resource consists of the inside-outside algorithm that was used for the training procedure, sustaining probabilistic context-free grammars.

The obtained models were evaluated on a test corpus. The results of our experiments show that PCFGs are good in predicting syllable boundaries, but simple PCFGs do not yield good results for grapheme-to-phoneme conversion.

## 2 Syllabification

A syllabifier splits a sequence of phonemes to syllables, e.g. the German phoneme sequence [lUftlOx] (*air pocket*) can be syllabified as [lU][ftlOx], [lUf][tlOx], [lUft][lOx], and [lUftl][Ox].

Our method, used for the experiments described in this paper, is based on a manually constructed context-free grammar with about 50 rules which returns for a given phoneme string all possible analyses. Our grammar describes how words are composed of syllables and syllables branch into onset, nucleus and coda. These syllable parts are re-written by the grammar as sequences of natural phone classes, e.g. stops, fricatives, nasals, liquids, as well as long and short vowels, and diphtongs. The phone classes are then re-interpreted as the individual phonemes that they are made up of. Figure 1 shows some of the rules of the context-free grammar.

The first rule (1.1) in figure 1 describes a

| | | |
|---|---|---|
| (1.1) Word | → | Syl |
| (1.2) Word | → | Syl Syl |
| (1.3) Word | → | Syl Syl Syl |
| (1.4) Syl | → | Nucleus |
| (1.5) Syl | → | Onset Nucleus |
| (1.6) Syl | → | Onset Nucleus Coda |
| (1.7) Syl | → | Nucleus Coda |
| (1.8) Onset | → | On |
| (1.9) Onset | → | On On |
| (1.10) On | → | Liquid |
| (1.11) Nucleus | → | Vow |
| (1.12) Vow | → | SVowel |
| (1.13) Coda | → | Off |
| (1.14) Coda | → | Off Off |
| (1.15) Off | → | Plosiv |
| (1.16) Off | → | Fricative |
| (1.17) Liquid | → | [l] |
| (1.18) Fricative | → | [f] |
| (1.19) Fricative | → | [x] |
| (1.20) Plosiv | → | [t] |
| (1.21) SVowel | → | [U] |
| (1.22) SVowel | → | [O] |

Figure 1: Fragment of a context-free syllable grammar

monosyllabic word and rule (1.2) and (1.3) a word consisting of two and three syllables, respectively. The subsequent rule (1.4) specifies a syllable without an onset and coda, whereas in rule (1.5) the coda is missing and in (1.7) the onset. Rule (1.6) depicts how a syllable consists of an onset, nucleus and coda. The next two rules (1.8)-(1.9) describe the complexity of the onset and the rules (1.13)-(1.14) the complexity of the coda, each consisting of one or two phonemes. An onsets consists of a liquid, which is shown in rules (1.10). Rule (1.12) describe a short vowel, which is re-written in rule (1.21)-(1.22) as the vowels [U] and [O]. According to rules (1.15)-(1.20) a coda consonant can be reinterpreted as a plosive or a fricative, which are in turn re-written as a [p], a [f], a [x], or a [t].

Figure 2 depicts one of four possible analyses of the phoneme string [lUftlOx].

We transform the context-free grammar by a training procedure to a probabilistic CFG. We then choose the analysis with the highest probability. The probability of one analysis is defined as the product of the probabilities of the grammar rules appearing in the analysis. In our example the correct syllable segmentation received the highest probability: [lUft][lOx].

We use the inside-outside algorithm devel-

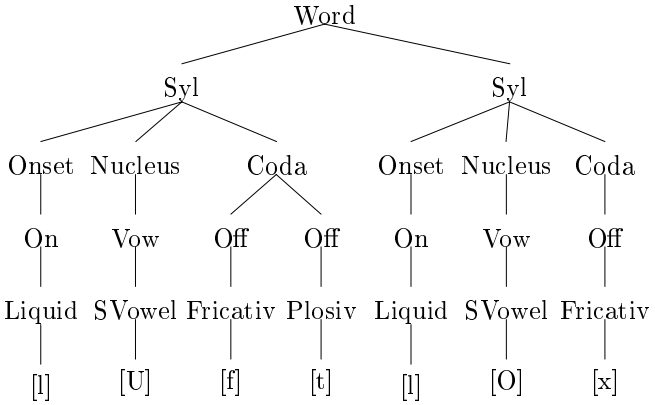


Figure 2: Possible segmentation of the word "Luftloch" *air pocket*

oped by Baker (1979), and generalized by Lari and Young (1990), for the transformation of a context-free grammar to a PCFG, the so called training procedure. In an initializing phase, the grammar rules are assigned random probabilities, which are reestimated during several iterations yielding the rule probabilities. There are three free parameters that can be varied: (1) the training corpus, (2) the number of iterations, and (3) the start parameters. We used the freely available lopar parser, implemented by Schmid (2000).

Figure 3 shows a fragment of the best performing PCFG with the rule probabilities used for syllabification. Rules (2.1)-(2.3) show that the most probable word structure is a word consisting of one syllable, a two-syllabic word is less probable and the least probable structure is a three-syllabic word. Almost 50% of the syllables consists of onset, nucleus and coda (rule (2.4)). Rules (2.5)-(2.6) show that syllables with empty onsets are preferred to open syllables. Simple onsets consisting of one consonant are more probable than complex ones, which is also true for codas (rules (2.7)-(2.8) and (2.13)-(2.14)). Rules (2.9)-(2.10) show that fricatives are more probable than liquids in the onset. Moreover, it is more likely that a nasal appears in the coda than a liquid (rule (2.15)-(2.16)).

### 2.1 Experiments

Our experiments are based on two different corpora: (i) a spoken news corpus of 1.5 h and (ii) the Sternzeit corpus, a feature series consisting of 2 hours read text. The corpus comprise

| | |
|---|---|
| (2.1) 0.450488 | Word → Syl |
| (2.2) 0.29816 | Word → Syl Syl |
| (2.3) 0.141141 | Word → Syl Syl Syl |
| (2.4) 0.487529 | Syl → Onset Nucleus Coda |
| (2.5) 0.198562 | Syl → Onset Nucleus |
| (2.6) 0.260016 | Syl → Nucleus Coda |
| (2.7) 0.878341 | Onset → On |
| (2.8) 0.115892 | Onset → On On |
| (2.9) 0.114081 | On → Liquid |
| (2.10) 0.234265 | On → Fricative |
| (2.11) 0.906921 | Nucleus → Vow |
| (2.12) 0.679005 | Vow → SVowel |
| (2.13) 0.854933 | Coda → Off |
| (2.14) 0.13788 | Coda → Off Off' |
| (2.15) 0.37867 | Off → Nasal |
| (2.16) 0.257495 | Off → Liq' |

Figure 3: Grammar fragment of the PCFG used for syllabification

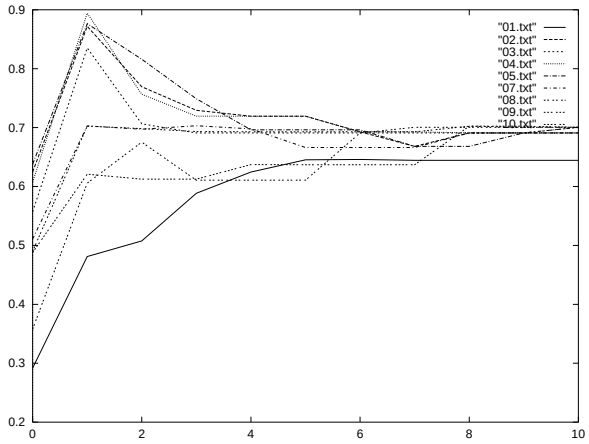


Figure 4: Accuracy of the syllabifier, number of iterations (x-axis), accuracy (y-axis)

96165 words, which are automatically looked up in a pronunciation dictionary, CELEX (Baayen et al., 1993) . The transcribed words are devided into 9/10 training and 1/10 test corpus. The training corpus does not include syllable boundaries, whereas the test corpus is annotated with syllable boundaries.

**Training.** We utilized the following training procedure:

- we initialize the CFG 10 times with randomized rule probabilities (10 start grammars),
- each of the start grammars is re-estimated 10 times on the training corpus with the inside-outside algorithm.

**Evaluation.** We evaluated our system on a test set of almost 9000 words. The ambiguity expressed as the average number of analyses per word was about 90. We evaluated the obtained models after each training iteration by calculating the most probable analysis (viterbi-parse) and extracting the syllable boundaries from the analysis. The yielded syllabified phoneme strings are compared with the annotated evaluation corpus and the syllable accuracy is measured, which is calculated as the number of the correctly predicted syllable boundaries devided by the number of all syllable boundaries. The results are shown in figure 4.

Each accuracy curve show how the current model performs on the test data. At the value 0 of the x-axis, the accuracy of the randomly initialized grammar is depicted. At the value 1, the accuracy after the first iteration, and at value 2 the accuracy after the second iteration is displayed, and so forth. The curves point out that five iterations would be enough for the syllabification task. Furthermore, it is quite important to experiment with a high number of start grammars. The accuracy usually increases of about 30% until the maxima is reached, independently which start grammar was used. The range between the best and the worst grammar of the start grammars is about 30%. The large range indicates that it is worthwhile to search for a start grammar with a high start precision value. The described training procedure offers a number of grammars that vary in their performance. The next step is to choose the best grammar with the highest performance, which yield a syllable accuracy of almost 90% on the test data.

**Qualitative Evaluation.** We found 938 errors that were made on the evaluation suite by the best model. The trained model showed main deficits in predicting the coda structure. 60.1% (i.e. 564) of the errors were made when the model predicts that the consonants belongs to the onset, however they are part of the coda. 39.9% (i.e. 374) of the errors were made when the models predicts the onset structure.

83% of the wrong coda prediction are made with alveolar consonants. Out of 564 items, 364 items were [*t*], followed by [*n*] with 42, [*ts*] with

34, [s] with 29, and [R] with 25 errors.

Fricatives seemed to be the main source of wrong onset prediction, almost 65% of the 374 errors are due to fricatives: [S] 149 errors, [s] 60 errors, [f] 34 errors, followed by [k] (28), [g] 25, [R] 24, [l] 18, [b] 15, [t] 12. The rest the errors of the following list occured less than 10 times: [p, d, m, ks].

If we analyze how the errors could be avoided, the most important improvement could be made by writing rules for prefixes and suffixes in the grammar similar to those suggested by Meng (2001) for English. 189 errors were made on prefixes and 108 on suffixes. A further improvement could be made by modelling numerals seperately in a pre-processing procedure, which could avoid 120 errors. Names and acronyms are found to be a minor source of errors (27, and 19 respectively). Another problem are morpheme boundaries in compound words. It would be very interesting to insert in the hierarchy a morphological level, where words are split to prefix, root and suffix, and where a recursive rule allows this again, as compounding in German is a very productive process.

**Evaluation of the perplexitiy.** The perplexity of a PCFG is measured after each training iteration on the training corpus and is defined as a monotonously decreasing function. Moreover, the inside-outside algorithm tries to reduce the perplexity of a PCFG during the training procedure. It is very interesting if the perplexity correlates with the accuracy. We extracted for each of the 10 start grammars the grammar with the best accuracy values and the perplexity measured at that point. The upper curve of figure 5 shows the best precision values of all 10 grammars and the lower one shows the measured perplexity. The grammars were ordered on the x-axis by decreasing accuracy. Figure 5 shows that there is no correlation between the accuracy and the perplexity.

## 3 Grapheme-to-phoneme conversion

A grapheme-to-phoneme converter transforms an orthographic word to its transcription, e.g. the German word Luftloch *(air pocket)* to the phoneme string [lUftlOx]. The aim of our work is to use the same grammar described in section 2. We enriched the grammar by rules that expand the phonemes to graphemes. Figure 6

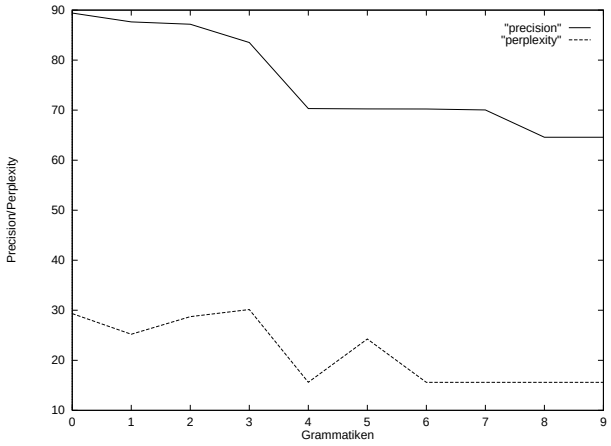


Figure 5: Syllabifier: grammars with maximal accuracy and their perplexity values

| | | |
|---|---|---|
| (3.1) phon=l | → | l |
| (3.2) phon=O | → | o |
| (3.3) phon=U | → | u |
| (3.4) phon=f | → | f |
| (3.5) phon=x | → | c h |
| (3.6) phon=t | → | t |

Figure 6: Fragment of the additional rules

shows a fragment of the expanded grammar. The example word Luftloch *(air pocket)* yields 41 analyses according to the new grammar. Figure 7 depicts the correct analysis. The main idea is to use the standard probability model for disambiguation of the analyses.

### 3.1 Experiments

In this section, we present the application of the standard probability model to grapheme-to-phoneme (G2P) conversion (i) using a CFG to produce all possible phonemic correspondences of a given grapheme string, (ii) predicting pronunciation by choosing the most probable analysis, and (iii) reading off the transcription from the phoneme tier. A fragment of the grammar is already described in section 2. We employed 389000 words of the Stuttgarter Zeitungskorpus (STZ), a German newspaper corpus for our training procedure.

**Training.** We used the same training procedure like in section 2.1:

- we initialize the CFG 10 times with randomized rule probabilities (10 start gram-

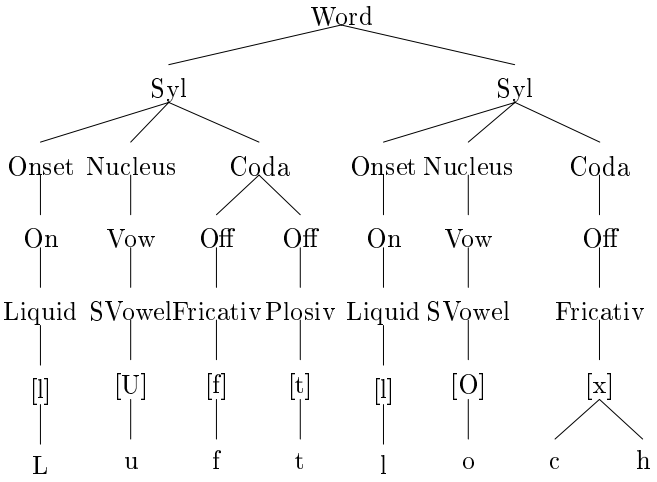


Figure 7: The correct grapheme-to-phoneme conversion of the word Luftloch *(air pocket)*

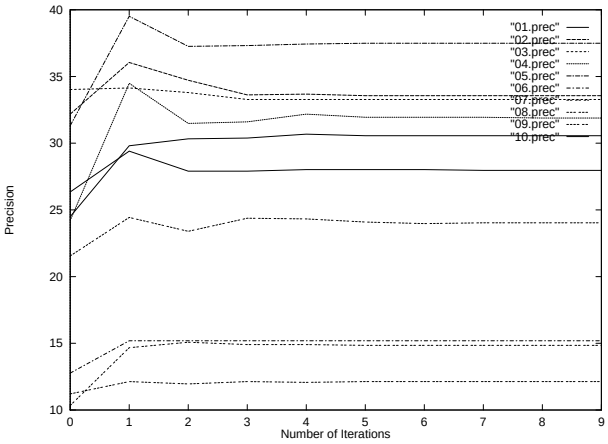


Figure 8: G2P accuracy of 10 start grammars over 10 training interations

mars),

- each of the start grammars is re-estimated 10 times on the training corpus with the inside-outside algorithm.

**Evaluation.** The evaluation corpus consists of 1731 words, not appearing in the training corpus. The words were extracted from 295105 words of the CELEX dictionary not appearing in the newspaper corpus. From this test set we manually eliminated (i) foreign words, (ii) acronyms, (iii) proper names, (iv) verbs, and (v) words that did not exactly consist of two syllables. The ambiguity expressed as the average number of analyses per word was 289. Each of the grammars is evaluated linguistically by comparing the most probable transcription of a word with the transcription of the CELEX dictionary. The word accuracy was measured by: the number of correctly analyzed words devided by the number of all words appearing in the test corpus. Figure 8 shows the results of the training procedure. Almost all of the 10 grammars reached the maximum of the accuracy after one iteration. The best grammar yields a word accuracy of almost 40%, which is nonsatisfying. The worse result can be due to the various parameters that play a role with PCFGs, e.g. size of the training corpus, number of start grammars. We systematically varied these parameters in additional experiments, presented in section 3.2

**Qualitative evaluation.** We found 1346 errors (in 1047 words) in the G2P task. Note, that there could be several errors in a word. Predicting vowel quality was a main error source, however it was not easy to find main error sources for consonants. 906 errors (i.e. 67.3%) are due to vowel quality. There are 641 errors where the algorithm predicted a short vowel instead of a long vowel: [$a$] instead of [$a$ :] - 179 times, [$U$] instead of [$u$ :] 125 times, [$O$] instead of [$o$ :] 110 times, [$I$] instead of [$i$ :] 102 times, [@] instead of [$e$ :] 54 times, [9] instead of [2 :] 30 times, [$Y$] instead of [$y$ :] 26 times, [$E$] instead of [$E$ :] 15 times. A further problem was to decide whether a /e/ is transcribed as a schwa [@] or a [$E$]. In 200 cases the model predicted that /e/ is transcribed as [@] instead of [$E$]. Some problems appeared when two adjacent identical graphemes were found e.g. Schneeball (*snow ball*). Between the two /ee/s, there could either be a morpheme boundary i.e. the first vowel belongs to the first syllable and the second vowel to the succeeding syllable, or the two vowels refer to a long vowel ([$e$ :]).

The errors regarding the consonants are multifarious. 155 out of 440 errors are made on the feature voiced vs. unvoiced. The model predicted a [$d$] instead of a [$t$] in 53 cases. [$p$] instead of [$b$] was predicted 11 times, and [$b$] instead of [$p$] 12 times. 39 times a [$v$] was predicted instead of a [$f$], and 20 times a [$s$] instead of a [$z$], and 12 times a [$z$] instead of a [$s$]. Another problem was that the algorithm transcribed 66 times a /s/ as a [$S$] in the coda, which can be avoided if the rule

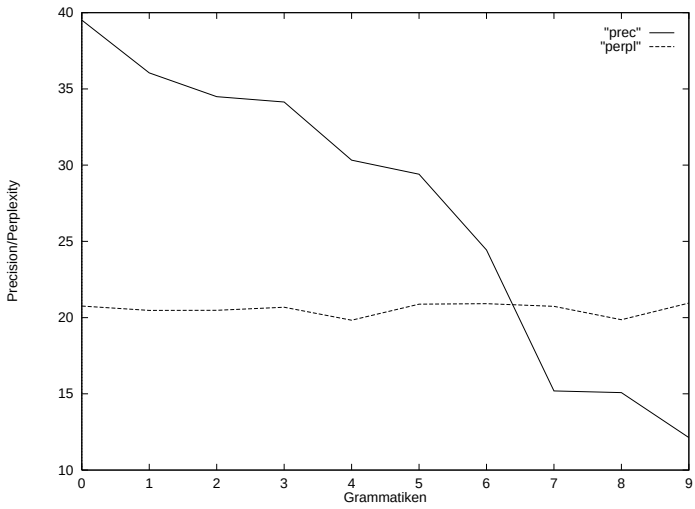


Figure 9: G2P: grammars with maximal accuracy and their perplexity value

$s \rightarrow S$ is restricted to the onset. A further error comes from the suffix /ig/ e.g. in flüssig (*liquid*). The model transcribed a /g/ 62 times as a $[x]$ [1] instead of a $[k]$. Another source of error (48 times) is that the algorithm transcribes an /n/ preceding a /k/ as an $[n]$ instead of a $[N]$. This rule have to be applied except when there is a morpheme boundary. Meng (2001) shows that for grapheme-to-phoneme conversion the modelling of prefixes and suffixes in the grammar could help to improve the performance of the trained model.

**Evaluation of the perplexity.** Figure 9 shows the results of the accuracy and the perplexity. The accuracy is a decreasing function, whereas the perplexity did not change remarkably. Thus, there is no correlation between accuracy and perplexity. The results correspond to the finding shown in section 2.1 for syllabification.

### 3.2 Additional experiments

**Start parameters.** We varied the parameter "start grammar", as the grammars are randomly initialized in the beginning of the training procedure and the inside-outside algorithm can only detect local maxima. We experimented with 50 randomly initialized start grammars yielding a 3% increase in accuracy to 42.5%.

**Size of training corpus.** In further experiments we varied the size of the training corpus systematically: 4500, 9600, 15000, 33000, 77000, 182000, 398000 and 1000000 words. We initialized 50 start grammars and trained each grammar with 10 iterations on the different corpora. The best grammar achieved a 42.6% accuracy on a corpus size of 182000 words. It is quite interesting that a corpus size of 398000 and 1000000 did not yield better results than a smaller corpus.

**Type training.** In another experiment we investigate if the accuracy can be increased by using a typenized training corpus, i.e. a training corpus where a word appears only once. The size of the training corpus was systematically varied: 250, 500, 1000, 2000, 4000, 8000, 16000, 32000 and 64000. We trained 50 randomly initialized start grammars with 10 training iterations. The accuracy increased from 38.01% with 250 types to 41.25% with 32000 types and then started to decrease again.

## 4 Discussion

We have presented an approach to unsupervised learning and automatic detection of syllable boundaries and grapheme-to-phoneme conversion using the standard probability model. Automatic conversion of a string of characters, i.e. a word, into a string of phonemes, i.e. its pronunciation, is essential for applications such as speech synthesis from unrestricted text input, which can be expected to contain words that are not in the system's pronunciation dictionary or otherwise unknown to the system. The phoneme string received by grapheme-to-phoneme conversion has to be syllabified before it can be further processed to speech. In our first experiment a phoneme string was segmented to syllables. The best model achieved a syllable accuracy of almost 90%. In a further experiment we added supplemantary grapheme-to-phoneme rules to the context-free grammar and applied the CFG to G2P. The results of 42.5% show that the G2P task cannot be solved sufficiently with a simple PCFG. The variation of the parameters: size of the training corpus, number of start grammars, and type training did not noteworthy increase the word accuracy.

We assume that the grammar models syllab-

[1] Note, that CELEX transcri e oth the velar fricative $[x]$ and the palatal fricative $[C]$ as $[x]$. They suggest that the correct fricative can e chosen in a pre-lexical step.

ification quite well but grapheme-to-phoneme conversion needs a more elaborate grammar that expresses e.g. the position of the syllable in a word, a different treatment of onset and coda consonants, the position of the consonant in the consonant cluster. Meng (2001) experimented with grammar rules that model prefixes, suffixes and roots, which could improve the performance of the models. Alternatively, we suppose that another probability model performs better on the G2P task.

Although, it is difficult to compare the performance with other systems and methods, we want to refer to several approaches that examined the syllabification and grapheme-to-phoneme conversion task. Müller et al. (2000) showed that G2P yields a word accuracy of 75% using multidimensional clustering models. They used a CFG to generate all possible phonemic corresponcences of a given grapheme string and then applied a probabilistic syllable model predicting pronunciation by choosing the most probable analysis. The probabilistic syllable model was trained on a large transcribed database. The Bell Labs German TTS system (Möbius, 1999) performed at better than 94% word accuracy on our test set. This TTS system relies on an annotation of morphological structure for the words in its lexicon and it performs a morphological analysis of unknown words (Möbius, 1998); the pronunciation rules draw on this structural information. Meng (2001) reported a 69.3% word accuracy on English test data. However, she trained and evaluated on about 10.000 most frequent words appearing in the Brown Corpus. Damper et al. (1999) reported a 72% word accuracy on unaligned English data. Bouma (2000) achieved a 92.6% word accuracy for Dutch, using a 'lazy' training strategy on data aligned with the correct phoneme string, and a hand-crafted system that relied on a large set of rule templates and a many-to-one mapping of characters to graphemes preceding the actual G2P conversion.

Müller (to appear 2001) showed that the results for syllabification can be improved to 96.5% with a new algorithm combining the advantages of treebank and bracketed corpora training. Van den Bosch (1997) investigated the syllabification task with five inductive learning algorithms. He reported a generalisation error for words of 2.2% on English data. However, in German (as well as Dutch and Scandinavian languages) compounding by concatenating word forms is an extremely productive process. Thus, the syllabification task is much more difficult in German than in English. Daelemans and van den Bosch (1992) report a 96% accuracy on finding syllable boundaries for Dutch with a backpropagation learning algorithm. Vroomen et al. (1998) report a syllable boundary accuracy of 92.6% by measuring the sonority profile of syllables.

## 5 Conclusion

We have presented an approach to unsupervised learning and automatic detection of syllable boundaries and grapheme-to-phoneme conversion using the standard probability model. In our first experiment a phoneme string was segmented to syllables. We achieved a syllable accuracy of 90%. In a further experiment we added supplementary rules to the context-free grammar and applied the CFG to grapheme-to-phoneme conversion. The results of 42.5% show that the G2P task cannot be solved sufficiently. The variation of the parameters: size of the training corpus, number of start grammars, and type training did not noteworthy increase the word accuracy. A more elaborate grammar that models morphological structure, might increase the accuracy. Moreover, another probability model increases the performance of the task.

## References


Harald R. Baayen, Richard Piepenbrock, and H. van Rijn. 1993. The CELEX lexical database—Dutch, English, German. (Release 1)[CD-ROM]. Philadelphia, PA: Linguistic Data Consortium, Univ. Pennsylvania.

J. Baker. 1979. Trainable grammars for speech recognition. In Klatt D. and Wolf J., editors, *Speech Communciation Papers for the 97th Meeting of the Acoustical Soc iety of America*, pages 547–550.

Gosse Bouma. 2000. A finite state and data-oriented method for grapheme to phoneme conversion. In *Proc. 1st Conf. North American Chapter of the ACL (NAACL)*, Seattle, WA.

Walter Daelemans and Antal van den Bosch.

1992. Generalization performance of back-propagation learning on a syllabification task. In M.F.J. Drossaers and A Nijholt, editors, *Proceedings of TWLT3: Connectionism and Natural Language Processing*, pages 27–37, University of Twente.

Robert I. Damper, Yannick Marchand, Martin J. Adamson, and Kjell Gustafson. 1999. Evaluating the pronunciation component of text-to-speech systems for English: a performance comparison of different approaches. *Computer Speech and Language*, 13:155–176.

K. Lari and S.J. Young. 1990. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4:35–56.

Helen Meng. 2001. A hierarchical lexical representation for bi-directional spelling-to-pronunciation/pronunciation-to-spelling generation. *Speech Communication*, 33:213–239.

Bernd Möbius. 1998. Word and syllable models for German text-to-speech synthesis. In *Proc. 3rd ESCA Workshop on Speech Synthesis (Jenolan Caves)*, pages 59–64.

Bernd Möbius. 1999. The Bell Labs German text-to-speech system. *Computer Speech and Language*, 13:319–358.

Karin Müller, Bernd Möbius, and Detlef Prescher. 2000. Inducing probabilistic syllable classes using multivariate clustering. In *Proc. 38th Ann. Meeting of the ACL*, Hongkong, China.

Karin Müller. to appear 2001. Automatic detection of syllable boundaries combining the advantages of treebank and bracketed corpora training. In *Proc. 39th Ann. Meeting of the ACL*, Toulouse, France.

Helmut Schmid. 2000. LoPar. Design and Implementation. [http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/LoPar-en.html].

Richard Sproat, editor. 1998. *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach.* Kluwer Academic, Dordrecht.

Antal Van den Bosch. 1997. *Learning to Pronounce Written Words: A Study in Inductive Language Learning.* Ph.D. thesis, Univ. Maastricht, Maastricht, The Netherlands.

Jean Vroomen, Antal van den Bosch, and Beatrice de Gelder. 1998. A Connectionist Model for Bootstrap Learning of Syllabic Structure. 13:2/3:193–220.