Automatic Detection of Syllable Boundaries Combining the Advantages of Treebank and Bracketed Corpora Training

Karin Müller
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
Azenbergstrasse 12
D-70174 Stuttgart, Germany
karin.mueller@ims.uni-stuttgart.de

Abstract

An approach to automatic detection of syllable boundaries is presented. We demonstrate the use of several manually constructed grammars trained with a novel algorithm combining the advantages of treebank and bracketed corpora training. We investigate the effect of the training corpus size on the performance of our system. The evaluation shows that a hand-written grammar performs better on finding syllable boundaries than does a treebank grammar.

1 Introduction

In this paper we present an approach to supervised learning and automatic detection of syllable boundaries. The primary goal of the paper is to demonstrate that under certain conditions treebank and bracketed corpora training can be combined by exploiting the advantages of the two methods. Treebank training provides a method of unambiguous analyses whereas bracketed corpora training has the advantage that linguistic knowledge can be used to write linguistically motivated grammars.

In text-to-speech (TTS) systems, like those described in Sproat (1998), the correct pronunciation of unknown or novel words is one of the biggest problems. In many TTS systems large pronunciation dictionaries are used. However, the lexicons are finite and every natural language has productive word formation processes. The German language for example is known for its extensive use of compounds. A TTS system needs a module where the words converted from graphemes to phonemes are syllabified before they can be further processed to speech. The placement of the correct syllable boundary is essential for the application of phonological rules (Kahn, 1976; Blevins, 1995). Our approach offers a machine learning algorithm for predicting syllable boundaries.

Our method builds on two resources. The first resource is a series of context-free grammars (CFG) which are either constructed manually or extracted automatically (in the case of the treebank grammar) to predict syllable boundaries. The different grammars are described in section 4. The second resource is a novel algorithm that aims to combine the advantages of treebank and bracketed corpora training. The obtained probabilistic context-free grammars are evaluated on a test corpus. We also investigate the influence of the size of the training corpus on the performance of our system.

The evaluation shows that adding linguistic information to the grammars increases the accuracy of our models. For instance, we coded the knowledge that (i) consonants in the onset and coda are restricted in their distribution, and (ii) the position inside of the word plays an important role. Furthermore, linguistically motivated grammars only need a small size of training corpus to achieve high accuracy and even out-perform the treebank grammar trained on the largest training corpus.

The remainder of the paper is organized as follows. Section 2 refers to treebank training. In section 3 we introduce the combination of tree-

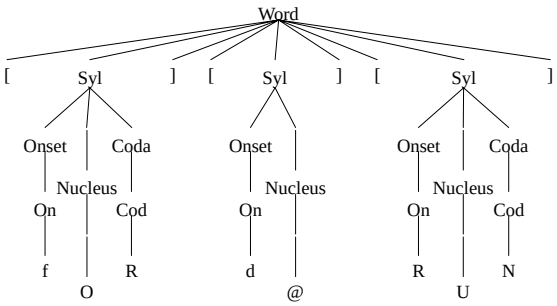


Figure 1: Example tree in the training phase

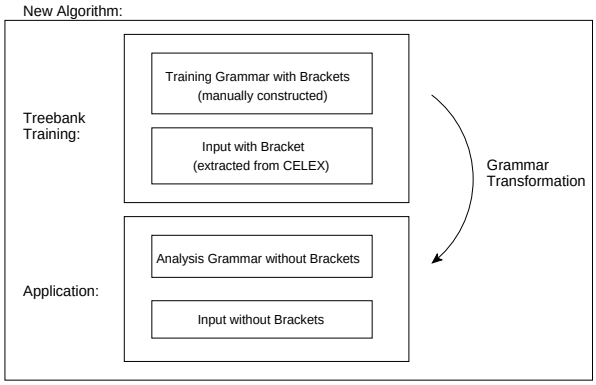


Figure 2: The novel algorithm that we capitalize on in this paper

bank and bracketed corpora training. In section 4 we describe the grammars and experiments for German data. Section 5 is dedicated to evaluation and in section 6 we discuss our results.

2 Treebank Training (TT) and Bracketed Corpora Training (BCT)

Treebank grammars are context-free grammars (CFG) that are directly read from production rules of a hand-parsed treebank. The probability of each rule is assigned by observing how often each rule was used in the training corpus, yielding a probabilistic context-free grammar. In syntax it is a commonly used method, e.g. Charniak (1996) extracted a treebank grammar from the Penn Wall Street Journal. The advantages of treebank training are the simple procedure, and the good results which are due to the fact that for each word that appears in the training corpus there is only one possible analysis. The disadvantage is that grammars which are read off a treebank are dependent on the quality of the treebank. There is no freedom of putting more information into the grammar.

Bracketed Corpora Training introduced by Pereira and Schabes (1992) employs a context-free grammar and a training corpus, which is partially tagged with brackets. The probability of a rule is inferred by an iterative training procedure with an extended version of the inside-outside algorithm. However, only those analyses are considered that meet the tagged brackets (here syllable brackets). Usually the context-free grammars generate more than one analysis. BCT reduces the large number of analyses. We utilize a special case of BCT where the number of analyses is always 1.

3 Combining the Advantages of TT and BCT

Our method used for the experiments is based on treebank training as well as bracketed corpora training. The main idea is that there are large pronunciation dictionaries that provide information about how words are transcribed and how they are syllabified. We want to exploit this linguistic knowledge that was put into these dictionaries. For our experiments we employ a pronunciation dictionary, CELEX (Baayen et al. (1993)) that provides syllable boundaries, our so-called treebank. We use the syllable boundaries as brackets. The advantage of BCT can be utilized: writing grammars using linguistic knowledge. With our method a special case of BCT is applied where the brackets in combination with a manually constructed grammar guarantee a single analysis in the training step with maximal linguistic information.

Figure 2 depicts our new algorithm. We manually construct different linguistically motivated context-free grammars with brackets marking the syllable boundaries. We start with a simple grammar and continue to add more linguistic information to the advanced grammars. The input of the grammars is a bracketed corpus that was extracted from the pronunciation dictionary CELEX. In a treebank training step we obtain a probabilistic context-free grammar (PCFG) by observing how often each rule was used in the training corpus. The brackets of the input guarantee an unambigous analysis of each word. Thus, we can apply the formula of treebank training given by (Char-

(1.1)	0.1774	Word → [Syl]
(1.2)	0.5107	Word → [Syl] [Syl]
(1.3)	0.1997	Word → [Syl] [Syl] [Syl]
(1.4)	0.4915	Syl → Onset Nucleus Coda
(1.5)	0.3591	Syl → Onset Nucleus
(1.6)	0.0716	Syl → Nucleus Coda
(1.7)	0.0776	Syl → Nucleus
(1.8)	0.9045	Onset → On
(1.9)	0.0918	Onset → On On
(1.10)	0.0036	Onset → On On On
(1.11)	0.0312	Nucleus → O
(1.12)	0.3286	Nucleus → @
(1.13)	0.0345	Nucleus → U
(1.14)	0.8295	Coda → Cod
(1.15)	0.1646	Coda → Cod Cod
(1.16)	0.0052	Coda → Cod Cod Cod
(1.17)	0.0472	On → f
(1.18)	0.0744	On → d
(1.19)	0.2087	Cod → R
(1.20)	0.0271	Cod → N

Figure 3: Grammar fragment after the training

niak, 1996): if r is a rule, let $|r|$ be the number of times r occurred in the parsed corpus and $\lambda(r)$ be the non-terminal that r expands, then the probability assigned to r is given by

$$p(r) = \frac{|r|}{\sum_{r' \in \{r' | \lambda(r') = \lambda(r)\}} |r'|}$$

We then transform the PCFG by dropping the brackets in the rules resulting in an analysis grammar. The bracketless analysis grammar is used for parsing the input without brackets; i.e., the phoneme strings are parsed and the syllable boundaries are extracted from the most probable parse. We want to exemplify our method by means of a *syllable structure grammar* and an exemplary phoneme string.

Grammar. We experimented with a series of grammars, which are described in details in section 4.2. In the following we will exemplify how the algorithm works. We chose the syllable structure grammar, which divides a syllable into onset, nucleus and coda. The nucleus is obligatory which can be either a vowel or a diphtong. All phonemes of a syllable that are on the left-hand side of the nucleus belong to the onset and the phonemes on the right-hand side pertain to the coda. The onset or the coda may be empty. The context-free grammar fragment in Figure 3 describes a so called *training grammar* with brackets.

We use the input word "Forderung" *(claim)* [fOR][d@][RUN] in the training step. The unambiguous analysis of the input word with the syllable structure grammar is shown in Figure 1.

Training. In the next step we train the context-free training grammar. Every grammar rule appearing in the grammar obtains a probability depending on the frequency of appearance in the training corpus, yielding a PCFG. A fragment [1] of the syllable structure grammar is shown in Figure 3 (with the recieved probabilities).

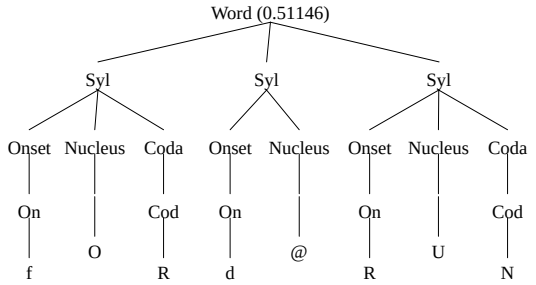
Rules (1.1)-(1.3) show that German disyllabic words are more probable than monosyllabic and trisyllabic words in the training corpus of 389000 words. If we look at the syllable structure, then it is more common that a syllable consists of an onset, nucleus, and coda than a syllable comprising the onset and nucleus; the least probable structure are syllables with an empty onset, and syllables with empty onset and empty coda. Rules (1.8)-(1.10) show that simple onsets are preferred over complex ones, which is also true for codas. Furthermore, the voiced stop [d] is more likely to appear in the onset than the voiceless fricative [f]. Rules (1.19)-(1.20) show the Coda consonants with descending probability: [R], [N].

Grammar transformation. In a further step we transform the obtained PCFG by dropping all syllable boundaries (brackets). Rules (1.4)-(1.20) do not change in the fragment of the syllable structure grammar. However, the rules (1.1)-(1.3) of the *analysis grammar* are affected by the transformation, e.g. the rule (1.2.) Word → [Syl] [Syl] would be transformed to (1.2.') Word → Syl Syl, dropping the brackets

Predicting syllable boundaries. Our system is now able to predict syllable boundaries with the transformed PCFG and a parser. The input of the system is a phoneme string without brackets. The phoneme string [fORd@RUN] (*claim*) gets the following possible syllabifications according to the syllable structure grammar: [fO][Rd@R][UN], [fO][Rd@][RUN], [fOR][d@R][UN], [fOR][d@][RUN], [fORd][@R][UN] and [fORd][@][RUN] .
The final step is to choose the most probable analysis. The subsequent tree depicts the most probable analysis: [fOR][d@][RUN], which is also the correct analysis with the overall word probability of 0.5114. The probability of one

[1] The grammar was trained on 389000 words

analysis is defined as the product of the probabilities of the grammar rules appearing in the analysis normalized by the sum of all analysis probabilities of the given word. The category "Syl" shows which phonemes belong to the syllable, it indicates the beginning and the end of a syllable. The syllable boundaries can be read off the tree: [fOR][d@][RUN].



4 Experiments

We experimented with a series of grammars: the first grammar, a *treebank grammar*, was automatically read from the corpus, which describes a syllable consisting of a phoneme sequence. There are no intermediate levels between the syllable and the phonemes. The second grammar is a *phoneme grammar* where only the number of phonemes is important. The third grammar is a *consonant-vowel grammar* with the linguistic information that there are consonants and vowels. The fourth grammar, a *syllable structure grammar* is enriched with the information that the consonant in the onset and coda are subject to certain restrictions. The last grammar is a *positional syllable structure grammar* which expresses that the consonants of the onset and coda are restricted according to the position inside of a word (e.g, initial, medial, final or monosyllabic). These grammars were trained on different sizes of corpora and then evaluated. In the following we first introduce the training procedure and then describe the grammars in details. In section 5 the evaluation of the system is described.

4.1 Training procedure

We use a part of a German newspaper corpus, the Stuttgarter Zeitung, consisting of 3 million words which are divided into 9/10 training and 1/10 test corpus. In a first step, we look up the words and their syllabification in a pronunciation dictionary. The words not appearing in the dictionary are discarded. Furthermore we want to examine the influence of the size of the training corpus on the results of the evaluation. Therefore, we split the training corpus into 9 corpora, where the size of the corpora increases logarithmically from 4500 to 2.1 million words. These samples of words serve as input to the training procedure.

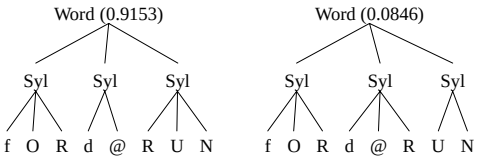
In a treebank training step we observe for each rule in the training grammar how often it is used for the training corpus. The grammar rules with their probabilities are transformed into the analysis grammar by discarding the syllable boundaries. The grammar is then used for predicting syllable boundaries in the test corpus.

4.2 Description of grammars

Treebank grammar. We started with an automatically generated treebank grammar. The grammar rules were read from a lexicon. The number of lexical entries ranged from 250 items to 64000 items. The grammars obtained start with 460 rules for the smallest training corpus, increasing to 6300 rules for the largest training corpus. The grammar describes that words are composed of syllables which consist of a string of phonemes or a single phoneme. The following table shows the frequencies of some of the rules of the analysis grammar that are required to analyze the word [fORd@RUN] (*claim*):

(3.1)	0.1329	Word	→	Syl Syl Syl
(3.2)	0.0012	Syl	→	f O R
(3.3)	0.0075	Syl	→	d @
(3.4)	0.0050	Syl	→	d @ R
(3.5)	0.0020	Syl	→	R U N
(3.6)	0.0002	Syl	→	U N

Rule (3.1) describes a word that branches to three syllables. The rules (3.2)-(3.6) depict that the syllables comprise different phoneme strings. For example, the word "Forderung" (*claim*) can result in the following two analyses:

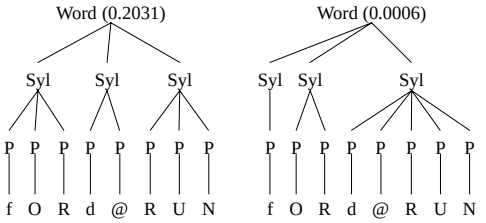


The right tree receives the overall probability of (0.0846) and the left tree (0.9153), which means that the word [fORd@RUN] would be syllabified: [fOR] [d@] [RUN] (which is the correct analysis).

Phoneme grammar. A second grammar is automatically generated where an abstract level is introduced. Every input phoneme is tagged with the phoneme label: P. A syllable consists of a phoneme sequence, which means that the number of phonemes and syllables is the decisive factor for calculating the probability of a word segmentation (into syllables). The following table shows a fragment of the analysis grammar with the rule frequencies. The grammar consists of 33 rules.

(4.1)	0.4423	Word	→	Syl Syl Syl
(4.2)	0.1506	Syl	→	P P
(4.3)	0.2231	Syl	→	P P P
(4.4)	0.0175	P	→	f
(4.5)	0.0175	P	→	O
(4.6)	0.0175	P	→	R

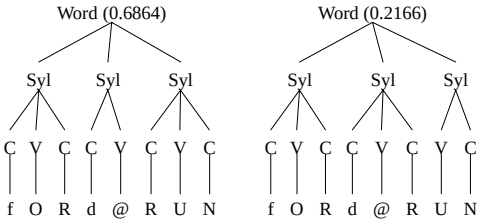
Rule (4.1) describes a three-syllabic word. The second and third rule describe that a three-phonemic syllable is preferred over two-phonemic syllables. Rules (4.4)-(4.6) show that P is re-written by the phonemes: [f], [O], and [R]. The word "Forderung" can be analyzed with the training grammar as follows (two examples out of 4375 possible analyses):



Consonant-vowel grammar. In comparison with the phoneme grammar, the consonant-vowel (CV) grammar describes a syllable as a consonant-vowel-consonant (CVC) sequence (Clements and Keyser, 1983). The linguistic knowledge that a syllable must contain a vowel is added to the CV grammar, which consists of 31 rules.

(5.1)	0.1608	Word	→	Syl
(5.2)	0.3363	Word	→	Syl Syl Syl
(5.3)	0.3385	Syl	→	C V
(5.4)	0.4048	Syl	→	C V C
(5.5)	0.0370	C	→	f
(5.6)	0.0370	C	→	R
(5.7)	0.0333	V	→	O
(5.8)	0.0333	V	→	@

Rule (5.1) shows that a three-syllabic word is more likely to appear than a mono-syllabic word (rule (5.2)). A CVC sequence is more probable than an open CV syllable. The rules (5.5)-(5.8) depict some consonants and vowels and their probability. The word "Forderung" can be analyzed as follows (two examples out of seven possible analyses):



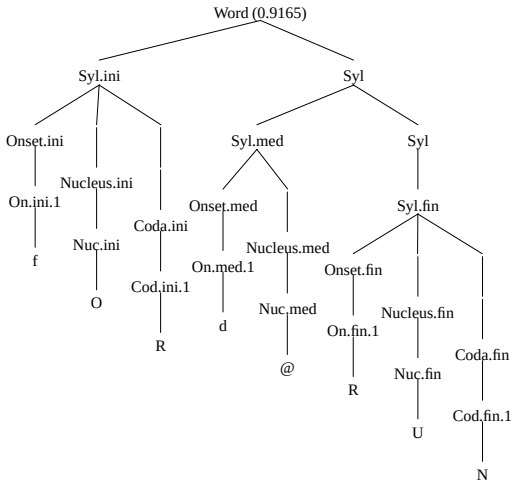
The correct analysis (left tree) is more probable than the wrong one (right tree).

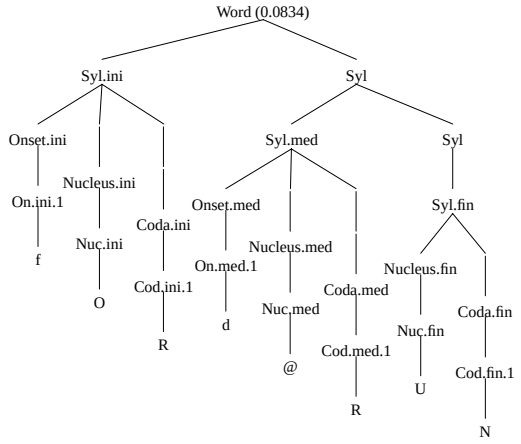
Syllable structure grammar. We added to the CV grammar the information that there is an onset, a nucleus and a coda. This means that the consonants in the onset and in the coda are assigned different weights. The grammar comprises 1025 rules. The grammar and an example tree was already introduced in section 3.

Positional syllable structure grammar. Further linguistic knowledge is added to the syllable structure grammar. The grammar differentiate between monosyllabic words, syllables that occur in inital, medial, and final position. Furthermore the syllable structure is defined recursively. Another difference to the simpler grammar versions is that the syllable is devided into *onset* and *rhyme*. It is common wisdom that there are restrictions inside the onset and the coda, which are the topic of phonotactics. These restrictions are language specific; e.g., the phoneme sequence [ld] is quite frequent in English codas but it never appears in English onsets. Thus the feature position of the phonemes in the onset and in the coda is coded in the grammar, that means for example that an onset cluster consisting of 3 phonemes are ordered by their position inside of the cluster, and their position inside of the word, e.g. On.ini.1 (first onset consonant in an initial syllable), On.ini.2, On.ini.3. A fragment of the analysis grammar is shown in the following table:

(6.1)	0.3076	Word → Syl.one
(6.2)	0.6923	Word → Syl.ini Syl
(6.3)	0.3662	Syl → Syl.fin
(6.4)	0.7190	Syl.one → Onset.one Rhyme.one
(6.5)	0.0219	Onset.one → On.one.1 On.one.2
(6.6)	0.0215	On.ini.1 → f
(6.7)	0.0689	Nucleus.ini → O
(6.8)	0.3088	Coda.ini → Cod.ini.1
(6.9)	0.0464	Cod.ini.1 → R

Rule (6.1) shows a monosyllabic word consisting of one syllable. The second and third rules describe a bisyllabic word comprising an initial and a final syllable. The monosyllabic feature "one" is inherited to the daughter nodes, here to the onset, nucleus and coda in rule (6.4). Rule (6.5) depicts an onset that branches into two onset parts in a monosyllabic word. The numbers represents the position inside the onset. The subsequent rule displays the phoneme [f] of an initial onset. In rule (6.7) the nucleus of an initial syllable consists of the phoneme [O]. Rule (6.8) means that the initial coda only comprises one consonant, which is re-written by rule (6.9) to a mono-phonemic coda which consists of the phoneme [R]. The first of the following two trees recieves a higher overall probabability than the second one. The correct analysis of the transcribed word /claim/ [fORd@RUN] can be extracted from the most probable tree: [fOR][d@][RUN]. Note, all other analyses of [fORd@RUN] are very unlikely to occur.





5 Evaluation

We split our corpus into a 9/10 training and a 1/10 test corpus resulting in an evaluation (test) corpus consisting of 242047 words.

Our test corpus is available on the World Wide Web[2]. There are two different features that characterize our test corpus: (i) the number of unknown words in the test corpus, (ii) and the number of words with a certain number of syllables. The proportion of the unknown words is depicted in Figure 4. The percentage of unknown words is almost 100% for the smallest training corpus, decreasing to about 5% for the largest training corpus. The "slow" decrease of the number of unknown words of the test corpus is due to both the high amount of test data (242047 items) and the "slightly" growing size of the training corpus. If the training corpus increases, the number of words that have not been seen before (unknown) in the test corpus decreases. Figure 4 shows the distribution of the number of syllables in the test corpus ranked by the number of syllables, which is a decreasing function. Almost 50% of the test corpus consists of monosyllabic words. If the number of syllables increases, the number of words decreases.

The test corpus without syllable boundaries, is processed by a parser (Schmid (2000)) and the probabilistic context-free grammars sustaining the most probable parse (Viterbi parse) of each word. We compare the results of the parsing step with our test corpus (annotated with syllable boundaries) and compute the accuracy. If the parser correctly predicts all syllable boundaries of

[2]http://www.ims.uni-stuttgart.de/phonetik/eval-syl

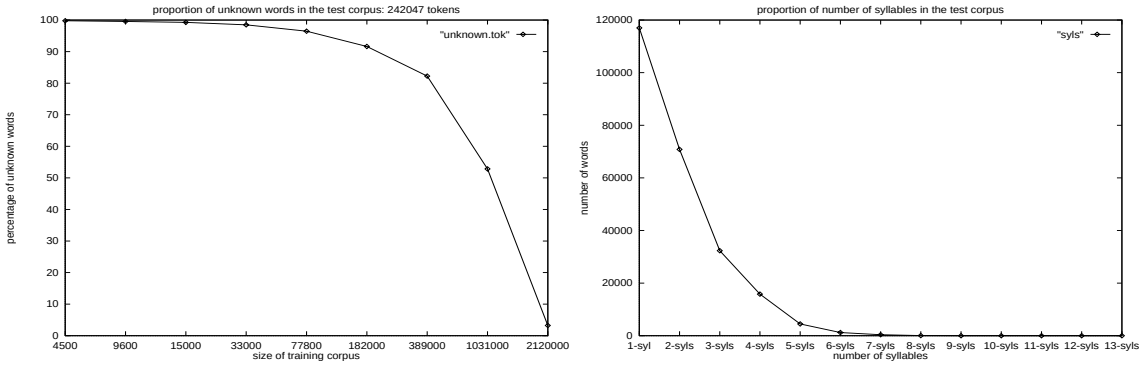


Figure 4: Unknown words in the test corpus(left); number of syllables in the test corpus (right)

grammars	word accuracy
treebank	94.89
phoneme	64.44
CV	93.52
syl structure	94.77
pos. syl structure	96.49

Figure 5: Best accuracy values of the series of grammars

a word, the accuracy increases. We measure the so called *word accuracy*.

The accuracy curves of all grammars are shown in Figure 6. Comparing the treebank grammar and the simplest linguistic grammar we see that the accuracy curve of the treebank grammar monotonically increases, whereas the phoneme grammar has almost constant accuracy values (63%). The figure also shows that the simplest grammar is better than the treebank grammar until the treebank grammar is trained with a corpus size of 77.800. The accuracy of both grammars is about 65% at that point. When the corpus size exceeds 77800, the performance of the treebank grammar is better than the simplest linguistic grammar. The best treebank grammar reaches a accuracy of 94.89%. The low accuracy rates of the treebank grammar trained on small corpora are due to the high number of syllables that have not been seen in the training procedure. Figure 6 shows that the CV grammar, the syllable structure grammar and the positional syllable structure grammar outperform the treebank grammar by at least 6% with the second largest training corpus of about 1 million words. When the corpus size is doubled, the accuracy of the treebank grammar is still 1.5% below the positional syllable structure grammar.

Moreover, the positional syllable structure grammar only needs a corpus size of 9600 to outperform the treebank grammar. Figure 5 is a summary of the best results of the different grammars on different corpora sizes.

6 Discussion

We presented an approach to supervised learning and automatic detection of syllable boundaries, combining the advantages of treebank and bracketed corpora training. The method exploits the advantages of BCT by using the brackets of a pronunciation dictionary resulting in an unambigous analysis. Furthermore, a manually constructed linguistic grammar admit the use of maximal linguistic knowledge. Moreover, the advantage of TT is exploited: a simple estimation procedure, and a definite analysis of a given phoneme string. Our approach yields high word accuracy with linguistically motivated grammars using small training corpora, in comparison with the treebank grammar. The more linguistic knowledge is added to the grammar, the higher the accuracy of the grammar is. The best model recieved a 96.4% word accuracy rate (which is a harder criterion than syllable accuracy).

Comparison of the performance with other systems is difficult: (i) hardly any quantitative syllabification performance data is available for German; (ii) comparisons across languages are hard to interpret; (iii) comparisons across different approaches require cautious interpretations. Nevertheless we want to refer to sev-

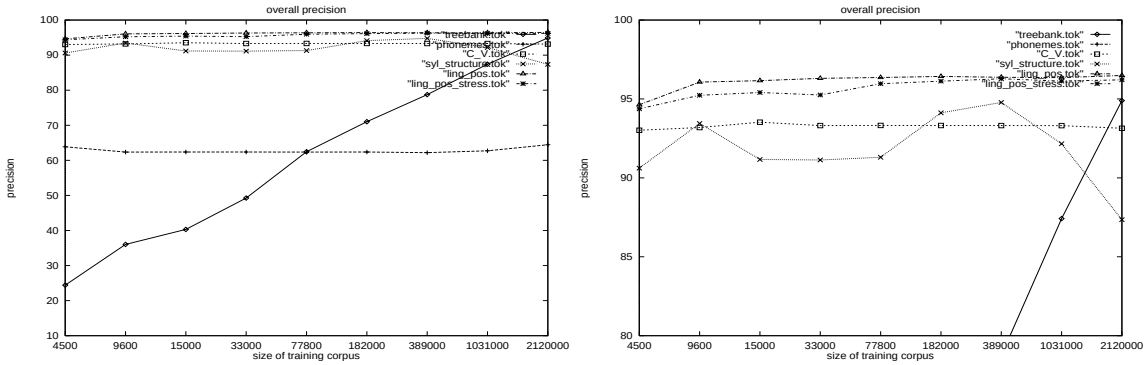


Figure 6: Evaluation of all grammars (left), zoom in (right)

eral approaches that examined the syllabification task. The most direct point of comparison is the method presented by Müller (to appear 2001). In one of her experiments, the standard probability model was applied to a syllabification task, yielding about 89.9% accuracy. However, syllable boundary accuracy is measured and not word accuracy. Van den Bosch (1997) investigated the syllabification task with five inductive learning algorithms. He reported a generalisation error for words of 2.22% 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. Future work is to apply our method to a variety of other languages.

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.

Juliette Blevins. 1995. The Syllable in Phonological Theory. In John A. Goldsmith, editor, *Handbook of Phonological Theory*, pages 206–244, Blackwell, Cambridge MA.

Eugene Charniak. 1996. Tree-bank grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI Press/MIT Press, Menlo Park.

George N Clements and Samuel Jay Keyser. 1983. *CV Phonology. A Generative Theory of the Syllable*. MIT Press, Cambridge, MA.

Walter Daelemans and Antal van den Bosch. 1992. Generalization performance of backpropagation 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.

Daniel Kahn. 1976. *Syllable-based Generalizations in English Phonology*. Ph.D. thesis, Massatchusetts Institute of Technology, MIT.

Karin Müller. to appear 2001. Probabilistic context-free grammars for syllabification and grapheme-to-phoneme conversion. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Pittsburgh, PA.

Fernando Pereira and Yves Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*.

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.