Improved Data–Driven Generation of Pronunciation Dictionaries Using an Adapted Word List¹

Matthias Wolff, Matthias Eichner and Rüdiger Hoffmann

Dresden University of Technology, Laboratory of Acoustics and Speech Communication
D-01062 Dresden, Germany
wolff|eichner|hoffmann@eakss2.et.tu-dresden.de

Abstract

Data-driven approaches to learning pronunciation variants for phonetic dictionaries have to deal with the problem of acquiring a sufficient amount of training data. The reason is not the size of the databases, but the unfavorable distribution of word frequencies in natural speech, which is known as Zipf’s law. In this paper we suggest a method which reorganizes a phonetic dictionary according to a given speech database in order to maximize the number of word models for which pronunciation variants can be learned with this corpus. Reorganization takes place automatically by analyzing the orthographic and phonetic transcriptions of the corpus. The method produces an alternative word list consisting of units ranging from partial words to multi-words. The efficiency and the limits of the approach are discussed on the basis of experiments carried out on the German VERBMOBIL corpus.

1. Introduction

Pronunciation dictionaries are the interface between phonetic and orthographic representation of speech. They are a substantial part of speech recognition systems and play an important role for the overall performance. Often canonical lexicons are manually extended by pronunciation variants to build a pronunciation dictionary. This is a time consuming process that depends on the skill and expert knowledge of the scientist. Therefore several attempts have been made to simplify the generation of pronunciation dictionaries (e.g. [1][2]).

In previous publications we introduced an algorithm for data – driven generation of pronunciation dictionaries from a speech data base [3][4]. Learning pronunciation variation from speech data means on the one hand determining a representative set of pronunciation variants for each word of the dictionary and a reliable estimation of their probability within the word model on the other. Previous dictionary learning experiments (see [3][5]) showed that the lack of training examples for most of the words in the word list is a basic problem.

In a typical spontaneous speech corpus the great majority of the words occur rarely (see Figure 1). This phenomenon is known as Zipf’s Law [6]. However, a reliable estimation of pronunciation variants and their probabilities requires a certain minimum number – at least 20 up to 30 – of examples per word [3] (This relatively low figure results from the efficient training strategy for word models introduced and discussed in [4]). Thus, for the example corpus in figure 1 only 13.7% of the word models can be trained.

Figure 1: Word frequency histogram of a typical spontaneous speech corpus.

This article introduces a method of reorganizing the word list which allows a much more efficient use of a given training database in terms of the number of word models for which reliable pronunciation variants can be learned.

2. Reorganizing the Word List

If the available training material is given, the only way to enhance its capabilities to train word models of a phonetic dictionary is to reorganize the dictionary itself. An obviously reasonable change is for instance to incorporate frequent multi-words (see e.g.[7][8]) or to split up numerals and compound words. However, the resulting parts must not be too small, since units of morpheme size or smaller are not usable as base units for the training of pronunciation variants (see e.g. [9][10]). So we focused on generating word units which occur frequently in the training corpus and which have a sufficient size (cf. [10]). The boundaries of the generated word units may, but do not necessarily have to, coincide with the natural word boundaries. Thus the word lists are not restricted to partial words, natural words and multi-words. Every frequent symbol string is considered to be a valid word unit (which of course is not optimal in view of the universality of the word...

¹ This work was partially founded by the German Federal Ministry of Education and Research (BMBF) under grant no. 524-4001-01 IV 701 L8 (VERBMOBIL). The responsibility for the text lies with the author.
list). Experiments showed, however, that the algorithm tends either to split words into components or to combine words to multi-words. Only a few cases of combinations of a word and a word component were observed.

We also implemented and tested more restrictive policies on the unit boundaries (see 2.2). This did not cause noticeable loss of performance but produced word lists that coincide better with standard dictionaries.

2.1. Prerequisites

The optimized word list for a given speech corpus is obtained by a statistical analysis of its orthographic and phonetic transcriptions. Any punctuation and special characters, including white spaces, are removed from the orthographic text. Word boundaries can, as an option, be marked and can serve as landmarks for the segmentation.

The automatic breakdown of a text corpus into word units requires a mapping of orthographic and phonetic symbol sets. This assures the persistence of the correct assignment of orthographic and phonetic text throughout the segmentation process. We map the symbol sets by a variable length joint orthographic and phonetic symbol set. We call symbols comprised of this combined alphabet graphones [10]. A graphone is defined as the combination of a minimal string of graphemes and a minimal string of phonemes. The notation \( g \) is used for graphones. \( g \) denotes a grapheme string and \( p \) denotes a phome string.

An algorithm which automatically derives the graphone alphabet from a pronunciation dictionary is described in [10].

<table>
<thead>
<tr>
<th>Novemberwoche</th>
<th>no:vEmb6vOx8</th>
</tr>
</thead>
<tbody>
<tr>
<td>n&lt;o:o1&gt;</td>
<td>v&lt;v</td>
</tr>
</tbody>
</table>

Figure 2: Example for graphone representation.

2.2. Algorithm

The suggested algorithm for reorganizing the word list has three stages:
- Compile orthographic and phonetic transcription of the speech corpus into the graphone representation. Optional: Insert word boundary symbols \(<_<>\).
- Make N-gram statistics on the succession of symbols in the graphone transcriptions. N-Grams from order 1 through to order 10 are built. Sentence boundaries are treated like word boundaries.
- Break down graphone transcription of the corpus into “word-like” units. “Word-like” means any graphone string of approximately the length of a conventional word. Segmentation takes place sentence by sentence. The obtained units are listed and counted.

2.2.1. Segmentation

The segmentation method bases on the evaluation of the conditioned interpolated multigram probability of each graphone symbol of the sentence and is described in [11]. The multigram probability can be calculated as follows:

\[
P(g_{1:n} | g_{1:n+1} = g_{1:n+1} | \cdots = g_{1:n+1}) = P(f)(h(g_{1:n}) + \sum_{i=2}^{n} F(i) h(g_{i:n+1} | g_{i:n} = g_{i:n+1})) \tag{1}
\]

The continuity of the multigram probabilities is described by the difference between two subsequent values:

\[
s_n = P(g_{1:n} | g_{1:n+1}) - P(g_{1:n+1} | g_{1:n+1}) \tag{2}
\]

There are great negative values of \( s_n \) at points where a symbol is improbable in relation to its history. This indicates the end of a frequently seen graphone sequence.

\[\text{Figure 3: Example for the segmentation of a German sentence. Upper diagram: conditioned interpolated multigram probability; lower diagram: difference value; dashed line: segmentation threshold } \zeta.\]

As the goal of the algorithm is breaking down the input into frequently seen graphone sequences, negative peaks of \( s_n \) are candidates for segment boundaries.

Segment boundary : \( s_n < \zeta \) \tag{3}

Criterion for a segment boundary is the shortfall of a threshold \( \zeta \) of the delta multigram probability \( s_n \).

2.2.2. Restrictions for Segment Boundaries

We investigated three different types of restricting the segment boundaries:
- Every word boundary represents a segment boundary. Additionally, segment boundaries within words are allowed. The resulting segments consist of words and partial words.
- Any location of a segment boundary is allowed. If a word boundary exists within a segment, the segment boundaries must lie on word boundaries. The resulting words are words, partial words and multi words [4].
- No restrictions to segment boundaries are applied. The segmentation process generates units of approximately the length of a word. Every segment is an entry in the optimized dictionary. For the use of such a dictionary in the recognition task a two step decoding algorithm is necessary. Firstly, a segment hypothesis graph is generated for the currently processed utterance. The second step consists of extracting the orthographic
representation from the segments and to map the resulting sequence on the orthographic representation of a canonical pronunciation dictionary. The best match is the decoded sequence of words.

2.3. Evaluation

The segmentation of a text database results in a list of used word units. Their absolute frequencies in the database are known from the N-gram statistic. Furthermore during the segmentation process, the actual frequency of usage is counted for each segment. The relative number \( R \) of those segments which occur at least \( \hat{H} \) times in the database is used as a quality measure for the segment list. Because short segments are not suitable as base units for the training of pronunciation variants (see above) this figure is discounted by the number of all segments which are shorter than a minimal length \( \hat{l} \).

\[
R = \frac{\sum_{seg} c_{seg}}{L}, \quad c_{seg} = \begin{cases} 1 & H_{seg} \geq \hat{H} \land l_{seg} \geq \hat{l} \\ 0 & \text{else} \end{cases}
\]

(4)

\( L \): number of different segments

\( H_{seg} \): absolute frequency of segment in database

\( l_{seg} \): length of segment (in graphone symbols)

The calculation of \( R \) assumes the usage of overlapping samples during the training of pronunciation variants. Overlapping means that for example \( n<\text{not}>i<\text{I}>c<\text{C}>t<\text{T}> \) ("not") will serve as a sample for the contained word \( i<\text{I}>c<\text{C}>t<\text{T}> \) ("I"). This seems to be reasonable in order to efficiently use training data. However, it has not yet been confirmed experimentally that the usage of overlapping samples does not compromise the quality of the learned pronunciation variants.

3. Experimental Results

The described method of generating an optimized word list for the data-driven training of pronunciation dictionaries was tested on material taken from the German VERBMOBIL II corpus. It consists of a text data base of 177,625 words (5,045 different).

The results shown below were obtained with a segmentation procedure which disregards the natural word boundaries (see 2.2.2). The example shown in figure 4 is taken from the same experiment. The coincidence with the natural word boundaries was not forced.

The delta multigram probabilities segmentation algorithm has two parameters the multigram weighting vector \( F \) and the segmentation threshold \( \zeta \). We investigated the influence of both parameters on the segmentation.

3.1.1. Weighting Vector \( F \)

<table>
<thead>
<tr>
<th>Weighting Vector</th>
<th>( \zeta )</th>
<th>( L )</th>
<th>( I = 1 )</th>
<th>( I = 3 )</th>
<th>( I = 4 )</th>
<th>( I_{seg} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL std. word list</td>
<td>-</td>
<td>20.3</td>
<td>17.6</td>
<td><strong>14.2</strong></td>
<td>4500</td>
<td>7.19</td>
</tr>
<tr>
<td>01 n/a (n-gram)</td>
<td>-0.15</td>
<td>27.7</td>
<td>25.6</td>
<td><strong>22.1</strong></td>
<td>14098</td>
<td>9.43</td>
</tr>
<tr>
<td>03</td>
<td>-0.15</td>
<td>37.3</td>
<td>33.0</td>
<td><strong>27.5</strong></td>
<td>8847</td>
<td>7.22</td>
</tr>
<tr>
<td>04</td>
<td>-0.15</td>
<td>41.9</td>
<td>33.4</td>
<td><strong>24.1</strong></td>
<td>4432</td>
<td>5.18</td>
</tr>
<tr>
<td>05</td>
<td>-0.15</td>
<td>40.4</td>
<td>30.5</td>
<td><strong>19.9</strong></td>
<td>3948</td>
<td>4.57</td>
</tr>
<tr>
<td>07</td>
<td>-0.15</td>
<td>28.3</td>
<td>23.4</td>
<td><strong>13.6</strong></td>
<td>4608</td>
<td>5.31</td>
</tr>
</tbody>
</table>

Table 1: Influence of the choice of weighting vectors on \( R \).

Table 2 shows the dependence of the relative number of “trainable” word models \( R \) (equation 4) on the choice of the
multigram weighting vector \( F \). The best values for \( R \) are achieved using vectors which prefer N-grams of higher order (line 03) or which weight all N-grams equally (line 04). The preference of short N-grams (lines 05 and 07) results in less (line 03) or which weight all N-grams equally (line 04). The achieved using vectors which prefer N-grams of higher order 3.1.2. Segmentation Threshold \( \zeta \)

Table 3 shows the influence of the segmentation threshold \( \zeta \) on \( R \). Choosing greater values of \( \zeta \) more segment boundaries will be detected and thus the average length of the segments will decrease. The best results were achieved for \( 0 > \zeta > -0.2 \).

<table>
<thead>
<tr>
<th>Weighting Vector</th>
<th>( \zeta )</th>
<th>( R(%) ) (( H = 30 ))</th>
<th>( L )</th>
<th>( \bar{l}_{seg} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL std. word list</td>
<td>-0.001</td>
<td>30.0</td>
<td>38.1</td>
<td>23.1</td>
</tr>
<tr>
<td>03</td>
<td>-0.02</td>
<td>45.5</td>
<td>37.2</td>
<td>26.4</td>
</tr>
<tr>
<td>03</td>
<td>-0.05</td>
<td>40.4</td>
<td>34.7</td>
<td>27.8</td>
</tr>
<tr>
<td>03</td>
<td>-0.15</td>
<td>37.7</td>
<td>33.0</td>
<td>27.5</td>
</tr>
<tr>
<td>03</td>
<td>-0.25</td>
<td>32.8</td>
<td>29.4</td>
<td>25.3</td>
</tr>
</tbody>
</table>

Table 2: Influence of segmentation threshold on \( R \).

By means of this method for optimizing word lists of speech corpora, the relative number \( R \) of word units for which pronunciation variants can be trained by a given corpus can be doubled compared to a standard word list (line 03 vs. line BL in Table 2).

4. Summary

We showed experimentally that by reorganizing the entries of a phonetic dictionary a given training corpus can be used much more efficiently for the automatic training of pronunciation variants. Using the obtained corpus-dependent word lists, consisting of partial words, words and multi-words, the number of word models that can be successfully trained with a given corpus was approximately doubled.

Experiments showed that the investigated multigram probabilities approach is suitable for reorganizing pronunciation dictionaries.

Our further research will focus on the following issues:
- Develop a method to estimate the parameters of the segmentation algorithm, i.e. the segmentation threshold and the weighting vector.
- Investigate the influence of using overlapping samples for dictionary training (see 2.3) on the consistency [3] of the word models.

5. References


