AUTOMATIC LEARNING OF FINITE STATE AUTOMATA FOR PRONUNCIATION MODELING.

M. Pastor, F. Casacuberta,

Institut Tecnològic d’Informàtica
Universitat Politècnica de València
Camí de Vera s/n 46071 València
{moises,fcn}@iti.upv.es

Abstract

The great variability of word pronunciations in spontaneous speech is one of the reasons for the low performance of the present speech recognition systems. The generation of dictionaries that take into account this variability can increase the robustness of such systems. A word pronunciation is a possible phone sequence that can appear in a real utterance, and represents a possible acoustic realization of the word.

In this paper, word pronunciations are modeled using finite state automata. The use of such models allow for the application of grammatical inference methods and an easy integration with the others sources of acknowledge. The training samples are obtained from the alignment between the phone decodification of each training utterance and the corresponding canonical transcription.

Models proposed in this work were applied in a translation-oriented speech task. The improvements achieved by these new models were in the range between 2.7 to 0.6 points depending on the language model used.

1. Introduction

In a speech recognition system, the mapping between words of the vocabulary and phone-like models are known as pronunciation models. Usual pronunciation models are sequences of phone-like models that correspond to the standard pronunciation that can be found in a dictionary (canonical pronunciation). The speech recognition systems based on such pronunciation models achieve good performance in a laboratory environment. However, the performance of such systems decreases dramatically in spontaneous environments. A spoken word should not present a large difference in pronunciation from its canonical representation to be understood by humans listeners. The human brain uses syntax, semantic and pragmatic knowledge to recover from partial information which is present in an utterance. Canonical models would be a good model for some words with enough acoustic information (long words). A small variation of a phone pronunciation does not represent an important part of the acoustic score of the word. Short words (pronouns, articles, etc.) are the most problematic ones, a small variation with respect to the canonical pronunciation can represent an important variation with respect to the canonical representation. These words are common in human language and usually do not have semantic information, and have a high n-gram probability of occurrence. This is way, for humans, an accurate pronunciation of these words is not important in order to be understood.

There are several approaches to automatic pronunciation modeling. One of the most interesting approaches is the phoneme-based rule-learning technique [10, 11]. The main problem that arises with these techniques is overgeneralization. On the other hand, one of their greatest advantages is their easy extension to infrequent or unobserved words. However, if a word is infrequent, the effect on the global performance of the system is small. For these words, we use only the canonical pronunciation as a model as is used for long words. Another alternative is to use finite state automata as pronunciation models [8]. The transitions of such automata are labelled by phone-like units. A path from the initial state to the final state represents a possible pronunciation of the word being modeled. One of the advantages of models of this type is the existence of a number of grammatical inference techniques to automatically learn such models from training pronunciation with different degrees of generalization [1, 3] and an easy integration with the rest of knowledge levels.

2. Automatic Learning of Word Pronunciations

2.1. Generation of the training pronunciations

Training samples are obtained from the alignment between the phone decodification of each training utterance and the corresponding canonical transcription. The alignment between two sequences of phone-like units is a subproduct of the computation of the editing distance between both sequences [12]. There can be equivalent (same distance) alignments. Our editing distance algo-
rithm gives a better score than those paths including substitutions and deletions because they are more probable than insertions in spontaneous speech [10].

For each word of vocabulary, \( w \in \Sigma \), let \( P_{\text{conv}}(w) \) be a set of pairs \( \{P^i_w, n^i_w\}; 1 \leq i \leq m(w) \) where \( P^i_w \) is the \( i \)th sequence of phones representing an acoustical realization of word \( w \), \( m(w) \) the number of different pronunciations for the word \( w \), and \( n^i_w \) the number of times that \( P^i_w \) is obtained from the alignments.

Firstly, it is necessary to select the words that will be modeled by their canonical pronunciations and those that will be modeled by grammatical inference. Currently, the criterion for choosing these words is their frequency in the corpus. In this work, we used a more restrictive criterion: we chose those words, \( w \) that have one or more pronunciations, \( P^i_w \) which appear(s) at least a given number of times \( \sigma \).

\[
\Theta_{\sigma}(\Sigma) = \{w | \exists(P^i_w, n^i_w) \in P_{\text{conv}}(w); n^i_w > \delta \} \quad (1)
\]

Note that the set defined by \( \Theta_{\sigma}(\Sigma) \) is contained in the corresponding set defined by the word frequency. In practice, long words do not appear in \( \Theta_{\sigma}(\Sigma) \).

The next step is to choose the pronunciations that are representative of a word. For a given word, \( w \in \Theta_{\sigma}(\Sigma) \), we take into account those pronunciations that appear at least a percentage of \( \delta \) of the total of pronunciations for word \( w \).

\[
\Gamma(w) = \{ (P^i_w, n^i_w) \in P_{\text{conv}}(w) | w \in \Theta(\Sigma) \land \frac{n^i_w}{\sum_j n^j_w} > \delta \} \quad (2)
\]

Due to the poor accuracy of phone decoders, some pronunciations are not suitable to be used for training. Those pronunciations that are far from the most representative and systematic pronunciations are automatically discarded.

Some examples of alternative pronunciation for some words are presented in Table 1.

### Table 1: Examples of alternative pronunciations for several Spanish words.

<table>
<thead>
<tr>
<th>Language</th>
<th>Pronunciations</th>
</tr>
</thead>
<tbody>
<tr>
<td>el</td>
<td>{el,44,(e,18),(l,17),(ol,11),(al,6), (en,5),(r,3),(on,3),(er,2),(ei,2)}</td>
</tr>
<tr>
<td>de</td>
<td>{(de,399),(d,41),(do,25),(da,24),(be,21), (e,15),(di,15),(o,5),(le,5),(@,5)}</td>
</tr>
<tr>
<td>favor</td>
<td>{(fabor,217),(fabo@,33),(fobor,12), (fibur,12), (fabur,12),(fabo,8),(fabr,6), (faba,5)}</td>
</tr>
<tr>
<td>por</td>
<td>{(por(220), (po,110), (pr,19), (pol,10), (or,7), (pu,3), (pur,2)}</td>
</tr>
<tr>
<td>una</td>
<td>{(na,116),(ona,83), (una,56), (ma,15), (ma,13), (ana,13), (@na,11), (gna,8), (hna,6),(bna,6,(ono,4)}</td>
</tr>
<tr>
<td>las</td>
<td>{(las,68), (los,20), (nas,11), (das,10), (uas,7),(nos,6), (dos,6),bos,5}</td>
</tr>
</tbody>
</table>

Three examples of the models inferred by the method proposed here are presented in Figures 1, 2 and 3. In figure 4, a canonical model for a long word (autobús) is also presented. The first three models presented here show a richer structure than the fourth.

### 3. Experimental Evaluation

#### 3.1. The TRAVELER Task Corpus

The general framework aimed at covering common sentences that are needed by a traveler visiting a foreign hotel. The scenario was limited to some human-to-human communication situations at a reception desk of a hotel: asking for rooms, wake-up calls, keys, the bill, moving the luggage, asking information about rooms, confirming a previous reservation, etc. For this purpose, a corpus was acquired during the first phase of the EuTrans project 1. The acoustic training subset corpus consists of the 1264 sentences by 16 speakers and the test data were composed of 336 sentences by 4 speakers. The different language models were trained with the transcriptions of the acoustic training subset. The utterances and speakers used to train the system were independent with respect to the ones used for testing. The vocabulary size of the Traveler task was 680 words. Test and training were gender balanced.

#### 3.2. System Overview

#### 3.3. Results

The recognition system is based on ATROS (Automatically Trainable Recognizer Of Speech) engine [4, 5, 6].

1 http://www.zeres.de/Eutrans/
6], ATROS is a continuous speech recognition system which uses stochastic finite-state models at all its levels: acoustic-phonetic, lexical and syntactic. All these models can be obtained in an automatic way [4]. This makes the system easily adaptable to different recognition tasks.

The acoustic front-end consists of frames of 25ms long with an interframe distance of 10ms. A filter bank of 21 trapezoidal filters with increasing widths according to the mel-frequency scale is applied to the 512-point FFT, producing 21 spectral weighted mean values. A discrete cosine transform is applied to these coefficients producing 10 mel-frequency cepstral coefficients. Energy is also added. First and second derivatives of cepstrum coefficients and energy complete the 33-component frame.

The acoustic models of phone-like units were 24 left-to-right continuous-density context-independent HMMs. They were trained with the HTK Toolkit [14]. The probability density functions of HMM states were modeled by Gaussian mixture densities with diagonal covariance matrices and were estimated with the standard Baum-Welch algorithm. Bigrams and trigrams used in these experiments were trained using the k-testable training algorithm [1, 2].

For decoding, the acoustic and pronunciation models are dynamically integrated in the syntactic model: the transitions in the syntactic model automaton are substituted by the corresponding pronunciation model, and each transition on the pronunciation model is substituted by the corresponding acoustic model (see Fig. 5). The decoding process is performed using the beam-search Viterbi algorithm [15] through the integrated network.

All experiments were done using a Pentium II 233 MHz with 64 Mb of memory, running Linux operating system. First of all, we determined the values for $\sigma$ and $\delta$. Zerograms and the training corpus Traveler were used in order to determine them. We tested the system for different language models and the test corpus using the best values obtained. The results presented in Table 2 showed better results when the new lexical models were used. The best performance improvement was obtained using bigrams with a 29.7 % word error rate reduction (see Table 2). The increment of the real time factor was not relevant due to the small size of increment (see Table 3).

### 4. Conclusions and Future Work

A method for automatically learning pronunciation models from speech data has been presented. The method has demonstrated that it improves the performance of the system. The greatest improvement in performance was obtained using bigrams, even when a narrow beam search was used. The Traveler corpus was a controlled corpus whose main goal was to train/test speech translation systems. Although the method proposed here allowed us to achieve better performances than with conventional ones, experiments with more spontaneous corpora will be performed.

### 5. Acknowledgements

This work was partially funded by the European EuTrans project (ESPRIT LTR-O30268) and by the Spanish project TAVAL (TIC-1FD1997-1433).
Table 2: Word error rate for different language models and different pronunciation models.

<table>
<thead>
<tr>
<th>LangMod</th>
<th>LexLin</th>
<th>LexAlt</th>
<th>Improv</th>
</tr>
</thead>
<tbody>
<tr>
<td>zero-gr</td>
<td>43.25</td>
<td>40.8</td>
<td>5.7</td>
</tr>
<tr>
<td>bigram</td>
<td>9.16</td>
<td>6.43</td>
<td>29.7</td>
</tr>
<tr>
<td>trigram</td>
<td>3.11</td>
<td>2.55</td>
<td>18.1</td>
</tr>
</tbody>
</table>

6. References


Table 3: Real Factor Time for different language models and different pronunciation models.

<table>
<thead>
<tr>
<th>LangMod</th>
<th>LexLin</th>
<th>LexAlt</th>
</tr>
</thead>
<tbody>
<tr>
<td>zero-gr</td>
<td>5.3</td>
<td>5.5</td>
</tr>
<tr>
<td>bigram</td>
<td>7.8</td>
<td>9.5</td>
</tr>
<tr>
<td>trigram</td>
<td>1.7</td>
<td>1.9</td>
</tr>
</tbody>
</table>


