STATISTICAL LEARNING OF LANGUAGE PRONUNCIATION STRUCTURE

Filipp Korkmazskiy

Multimedia Communications Research Lab,
Lucent Technologies Bell Labs, Murray Hill, NJ 07974, USA
yelena@research.bell-labs.com

ABSTRACT

This paper presents a new approach to rule based pronunciation generation. A system presented in this paper can automatically learn a new language pronunciation structure and to use this knowledge for pronunciation generation for an arbitrary context sensitive language. Unlike conventional text-to-speech systems which are based on the cost expensive human expert knowledge about specific language, this system can learn by using only a set of spellings and pronunciations. The pronunciations can be obtained either from a pronunciation dictionary or from a phonetically labeled database. The system ability to learn pronunciation structure for any context sensitive language makes it a valuable tool for development of multilingual speech recognition systems. In this study we present experimental results on automatic generation of pronunciations for English, German, Spanish, French and Italian languages.

1. INTRODUCTION

Formalization of phonological information is frequently an essential element in pronunciation modeling. Such formalization is required either for pronunciation generation of unknown words or for representation of some intraword phonological phenomena. The formalized phonological information can be presented in terms of pronunciation rules([1]), decision trees([2]), and artificial neural networks([3]). Formalization is very essential for developing of multilingual speech recognition systems. Development of such kind of systems often requires a knowledge about language specific pronunciation phenomena. A person who has to develop such kind of system usually does not have a full knowledge about all these languages. Pronunciation in such kind of systems usually is derived from the pronunciation dictionaries. Such approach cannot model pronunciations for the words which are not included in the pronunciation dictionary. Proper nouns(different personal names or names of the cities) may serve as an example of such a group of words. This problem can be solved by the use of language specific text-to-speech(TTS) systems which transform an arbitrary string of letters into a corresponding string of phonemes. TTS systems are based on formalized phonogical information usually represented in the form of the phonological rules built by a human expert. A development of such TTS systems requires human expert knowledge about specific language and it is usually time consuming and expensive. TTS systems are language specific and they can not be easily adapted to a new language. A system presented in this paper can *automatically learn a new language pronunciation structure* and to use this knowledge for pronunciation generation. There is no need to use a human expert knowledge and the same system can be used to *learn pronunciations rules for arbitrary context sensitive language*. A language specific pronunciation dictionary is the only input to this system. In addition to a pronunciation dictionary a phonetically labeled database can also be used to derive language specific pronunciation rules. This work represents a new development of our previous studies devoted to modeling pronunciation variations([4, 5, 6, 7]).

This paper is organized as follows. In Section 2, we define mutual information as a measure of association between letters and phonemes and show how this measure can be used to align letters and phonemes. Section 3 is devoted to a rule based pronunciation generation. It defines a structure of pronunciation rules and introduces a hierarchical rule derivation algorithm. In Section 4, experimental results on pronunciation generation for different languages are presented.

2. STATISTICAL ANALYSIS OF PRONUNCIATION STRUCTURE

2.1. Association between letters and phonemes.

Assume a set of words is represented by word spellings and word pronunciations. Word pronunciations can be obtained either from a dictionary or from a phonetically labeled database. There are also 2 alphabets: a letter alphabet and a phoneme alphabet. The letter alphabet consists of the letter symbols used in word spellings, the phoneme alphabet consists of the phoneme symbols used in word pronunciations. Our task is to evaluate a degree of association between each letter and each phoneme using some statistics accumulated over all pairs of spellings and pronunciations. In the absense of symbol alignment between spellings and pronunciations one of the way to estimate a degree of association between, say, letter A and phoneme B is to measure their *mutual information* $I(A, B)$ at word level. Assume $P(A)$ is probability of the word containing a letter A in the spelling, $P(B)$ is probability of the word containing a phoneme B in the pronunciation, and $P(A, B)$ is probability of the word containing the letter A in the spelling and the phoneme B in the pronunciation. Mutual information $I(A, B)$ between letter A and phoneme B is estimated as follows:

$$I(A,B) = \log\left[\frac{P(A,B)}{P(A)\cdot P(B)}\right] \tag{1}$$

Probability of the word containing some specified symbol(either a letter or a phoneme) is dependent on the word length(represented either as a number of letters in the spelling or as a number of phonemes in the pronunciation). For example, probability $P(B)$ of the word containing phoneme B is dependent on the length of pronunciation. The longer word pronunciation, the more probable it contains phoneme B. Similar property holds for probability $P(A)$ and the length of spelling. That is why we suggest that probability $P(A)$ should be conditioned by the length of spelling and probability $P(B)$ should be conditioned by the length of pronunciation. Let's also notice that a longer spelling length usually means a longer pronunciation length. To simplify our model we assume that both probabilities $P(A)$ and $P(B)$ should be conditioned only by a single length, say, the length of word pronunciation. Now, rather than using unconditional mutual information $I(A,B)$ we should introduce conditional mutual information $I(A,B|M)$ which is dependent on the length M of word pronunciation:

$$I(A,B|M) = \log\left[\frac{P(A,B|M)}{P(A|M)\cdot P(B|M)}\right] \tag{2}$$

Here, $P(A|M)$ is probability of the word containing a letter A, $P(B|M)$ is probability of the word containing a phoneme B, and $P(A,B|M)$ is probability of the word containing both a letter A and a phoneme B - all these probabilities are evaluated only for such words that have a length of pronunciation equals to M. To measure *a strength $g(A,B)$ of association* between letter A and phoneme B we propose to use an expectation of mutual information $I(A,B|M)$ over different values of M:

$$g(A,B) = \sum_M P(M)\cdot I(A,B|M) \tag{3}$$

Here $P(M)$ is a probability of word pronunciation length equals to M. A strength of association $g(A,B)$ should be evaluated for all pairs of letters and phonemes.

2.2. Letter-to-phoneme alignment.

The strength of association between letters and phonemes can be used to align letters in word spelling with corresponding phonemes in word pronunciation. This procedure can be implemented using dynamic programming(DP) algorithm. DP algorithm searches for such an alignment between letters in word spelling and phonemes in word pronunciation that a total accumulated association strength between all letters in the spelling and the corresponding phonemes in the pronunciation is maximized. Assume a word w has spelling $Sp(w) = \{A_1, A_2, \ldots, A_n, \ldots, A_N\}$, represented by N letters and pronunciation $Pr(w) = \{B_1, B_2, \ldots, B_m, \ldots, B_M\}$, represented by M phonemes. Letter-to-phoneme alignment has a following structure:

- Each letter A_n in the spelling $Sp(w)$ is aligned with a single phone or with a group of sequential phonemes in the pronunciation $Pr(w)$.
- Each phoneme B_m in the pronunciation $Pr(w)$ is aligned with a single letter or with a group of sequential letters in the spelling $Sp(w)$.

A total accumulated association strength $G(w)$ between all letters in the spelling $Sp(w)$ and corresponding phonemes in the pronunciation $Pr(w)$ is evaluated as follows:

$$G(w) = \sum_{n=1}^{N}\sum_{r=1}^{R(A_n)} g(A_n, B_r), \tag{4}$$

where $g(A_n, B_r)$ is association strength between letter A_n and phoneme B_r which is aligned with the letter A_n, $R(A_n)$ is a total number of the phonemes aligned with A_n. $G(w)$ is maximized by optimal segmentation of pronunciation $Pr(w)$ into the groups of phonemes aligned with corresponding letters of spelling $Sp(w)$.

Once the optimal alignment between spelling and pronunciation for all words is completed we are able to reevaluate association strength $g(A,B)$ defined in the section 2.1. A value of $P(A,B|M)$ was defined before as probability of the word containing a letter A in the spelling and a phoneme B in the pronunciation, provided that the length of pronunciation equals to M. No information about mutual alignment between A and B was used before for estimation of $P(A,B|M)$. Since this information becomes available $P(A,B|M)$ can be substituted by a probability $P_a(A,B|M)$ that accounts only for such words where the letter A is aligned with the phoneme B. This should change a value of mutual information(see f-la (2)) as follows:

$$I(A,B|M) = \log\left[\frac{P_a(A,B|M)}{P(A|M)\cdot P(B|M)}\right] \tag{5}$$

As a result reestimated values of association strength $g(A,B)$ (see f-la (3)) for all pairs (A,B) of letters and phonemes become more accurate since the new estimates incorporate alignment information(see f-la (5)).

Alignment between spellings and pronunciations can be repeated using these new more accurate values of association strength $g(A,B)$. And association strength can be again reevaluated by using the new alignment results. This process can be iterated until we achieve some stabilization in alignment.

3. RULE BASED PRONUNCIATION GENERATION

3.1. Pronunciation rule structure.

Once alignment is completed and mapping between letters and phonemes is established we can start looking for letter-to-phoneme rules that govern that mapping. Let's define *a pronunciation rule as a mapping $A \Rightarrow B$* between a letter A and a phoneme B. We assume that this mapping appears in the context C which represents letters adjacent to A and phonemes adjacent to B. Each such a context $C = (C_1, C_2, C_3)$ consists of the 3 contextual groups C_1, C_2 and C_3:

$$C_1 = (C_1^{(1)}, C_1^{(2)}, \ldots, C_1^{(N_1)})$$
$$C_2 = (C_2^{(1)}, C_2^{(2)}, \ldots, C_2^{(N_2)}) \quad (6)$$
$$C_3 = (C_3^{(1)}, C_3^{(2)}, \ldots, C_3^{(N_3)}).$$

Here C_1 is the left context of the letter A consisting of the N_1 letters, C_2 is the right context of the letter A consisting of the N_2 letters, C_3 is the left context of the phoneme B consisting of the N_3 phonemes. Such a structure of the context corresponds to left-to-right letter-to-phoneme generation process when left and right letter contexts and only left phoneme contexts are available. The mapping $A \Rightarrow B$ can be characterized by the probability of this mapping. The mapping $A \Rightarrow B$ may have different probability depending on the context C. The probability $P(A \Rightarrow B|C)$, can be evaluated as follows:

$$P(A \Rightarrow B|C) = \frac{M(A \Rightarrow B|C)}{M(A|C)} \quad (7)$$

Here $M(A \Rightarrow B|C)$ is a total number of the letter A to the phoneme B mappings provided that these mappings appear in the context C, and $M(A|C)$ is a total number of the letter A mappings to any phoneme provided that A appears in the context C.

3.2. Hierarchical rule derivation algorithm.

Pronunciation rule is a letter-to-phoneme mapping $A \Rightarrow B$ that appears in the context $C = (C_1, C_2, C_3)$. It is possible that some symbols in C carry very little or no information about $A \Rightarrow B$ mapping. The problem is to detect such low importance symbols and discard them while preserving the most important ones. This way we go from the higher hierarchy level rules to the lower hierarchy level rules. Let's define a *level of rule hierarchy* as a total number l of letters and phonemes in the rule context $C^{(l)}$. Letter-to-phoneme mapping probability $P(A \Rightarrow B|C^{(l)})$ can be defined for different levels l of rule hierarchy.

We can transform the l-level rule into the $(l-1)$-level rule, if the corresponding rule probabilities $P(A \Rightarrow B|C^{(l)})$ and $P(A \Rightarrow B|C^{(l-1)})$ are close enough to each other. Let's denote the absolute value of a difference between probabilities $P(A \Rightarrow B|C^{(l)})$ and $P(A \Rightarrow B|C^{(l-1)})$ as $\Delta P(A \Rightarrow B|C^{(l)} \leftrightarrow C^{(l-1)})$:

$$\Delta P(A \Rightarrow B|C^{(l)} \leftrightarrow C^{(l-1)}) =$$
$$= \mathbf{abs}[P(A \Rightarrow B|C^{(l)}) - P(A \Rightarrow B|C^{(l-1)}] \quad (8)$$

The l-level rule converts into the $(l-1)$ level rule by excluding one symbol, say c_m from the context $C^{(l)}$ of the l-level rule: $C^{(l)} = (c_1, c_2, \ldots, c_{m-1}, c_m, c_{m+1}, \ldots, c_l)$, $C^{(l-1)} = (c_1, c_2, \ldots, c_{m-1}, c_{m+1}, \ldots, c_{l-1})$
Let's describe algorithm for deriving multilevel system of rules:
Step 1. Collect all contexts $C^{(l)}$ $(l = L)$ for mapping $A \Rightarrow B$.
Step 2. Derive a set $S(l-1)$ of all contexts $C^{(l-1)}$ from all the contexts $C^{(l)}$ by excluding a single symbol c_m at the m-th position of the each context $C^{(l)}(1 \le m \le l)$.
Step 3. Select a context $C^{(l-1)} \in S(l-1)$ such that $C^{(l-1)}$ covers a maximum number $N(C^{(l-1)})$ of contexts $C^{(l)}$ at the l-th level and

$$\Delta P(A \Rightarrow B|C^{(l)} \leftrightarrow C^{(l-1)}) < \delta \quad (9)$$

for all such contexts $C^{(l)}$. In case of a tie give a preference to such a context $C^{(l-1)}$ that minimizes a maximum value of $\Delta P(A \Rightarrow B|C^{(l)} \leftrightarrow C^{(l-1)})$.
Step 4. Eliminate all contexts $C^{(l)}$ that are covered by the context $C^{(l-1)}$.
Step 5. Repeat steps 3-4 until a condition (9) holds for at least 1 rule $C^{(l)}$.
Step 6. Repeat steps 3-5 for $l = L-1, L-2, \ldots, 2$.
By the end of this procedure a multilevel hierarchical system of rules is created for the mapping $A \Rightarrow B$.

4. EXPERIMENTAL RESULTS

In order to generate pronunciation $P(w)$ for a word w we use its spelling $Sp(w)$ and the multilevel hierarchical system of rules described before. Dynamic programming implementation allows to select optimal letter-to-phone rule at each step of generation procedure. A logarithm of the rule probability $P(A \Rightarrow B|C)$ is used as a local distance measure in the DP search. In our experiments we used pronunciation dictionaries for Spanish, English, French, German and Italian languages. A size of the dictionaries varied from 20000-40000 words for German, Italian, French and Spanish to 100000 words for English. Approximately 80% of the dictionary words were used as a training set and the remaining 20% words as a test set. The accuracy of generated pronunciations was measured by counting a relative number of the rule-based generated pronunciations that were identical to the pronunciations generated by a corresponding TTS system. The results of automatic pronunciation generation for Spanish, English, French, German and Italian languages are presented for training and test data(Fig.1 – Fig.5).

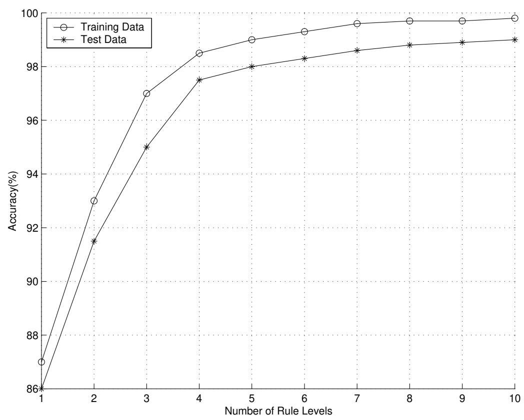

Figure 1: **Spanish Pronunciation.**

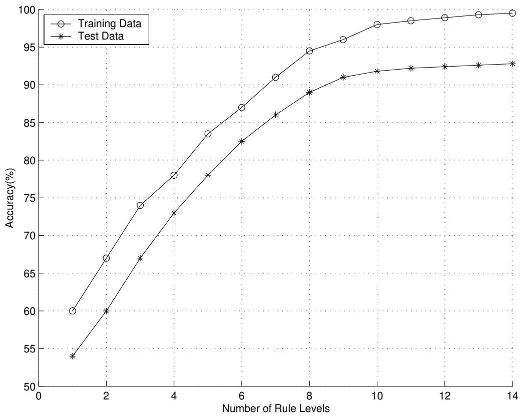

Figure 2: **English Pronunciation.**

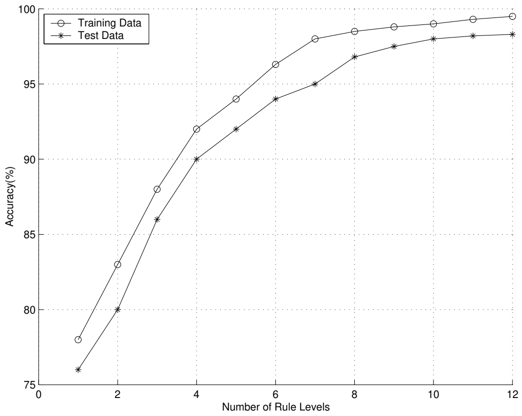

Figure 3: **French Pronunciation.**

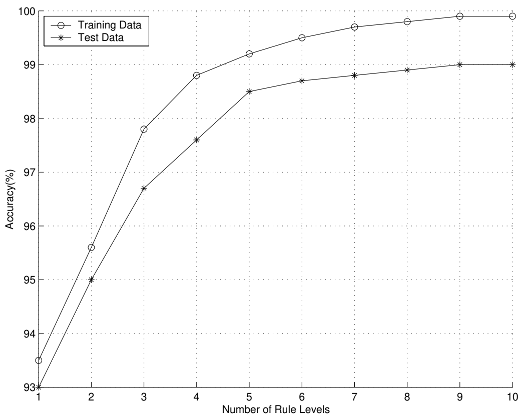

Figure 5: **Italian Pronunciation.**

5. CONCLUSIONS

In this paper we presented a method that can automatically learn language pronunciation structure for any context sensitive language. Pronunciation structure(rules) can be learned either from a pronunciation dictionary or from a

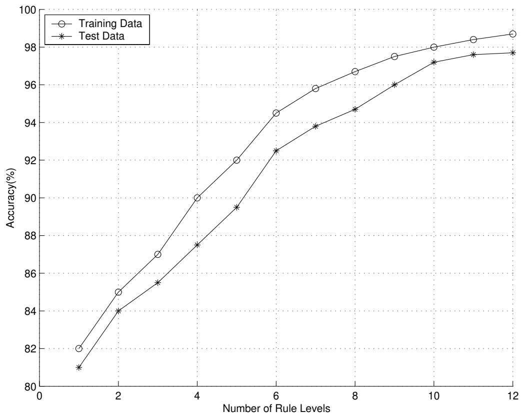

Figure 4: **German Pronunciation.**

phonetically labeled database. Complexity of learning pronunciation structure depends on language uniformity, i.e. similarity of pronunciations for different words. English is an example of the least uniformity. Languages like Spanish and Italian display the maximum uniformity. Proposed method can be expanded to learning pronunciation phenomena in spontaneous speech.

6. REFERENCES

[1] N. Gremelie , & J.-P. Martens, "In search of pronunciation rules," *Proc. Modeling Pronunciation Variation for Automatic Speech Recognition, Rolduc,* pp. 23-27, 1998.

[2] M. Riley, A. Ljolje, D. Hindle, and F. Pereira, "Automatic generation of detailed pronunciation lexicons" *In Automatic Speech and Speaker Recognition: Advanced Topics*, C.-H. Lee, F.K. Soong, and K.K. Paliwal, Eds. Kluwer Academic, Boston, March 1996, ch. 12.

[3] T. Fukada, T. Yoshimura, Y. Sagisaka, "Automatic generation of multiple pronunciation based on neural networks," *Speech communication,* **27(1)**, pp. 63-73, 1999.

[4] F. Korkmazskiy and C.-H. Lee, "Generating Alternative Pronunciations from a Dictionary," *Proc. Eurospeech99, Budapest*, 1999.

[5] F. Korkmazskiy C.-L.Shin and C.-H. Lee, "Pronunciation Modeling by Genetic Algorithm", *Proc. ASRU99, Kolorado,* pp. 58-63, 1999.

[6] I.A. Amdal, F.E. Korkmazskiy, A.C. Surendran, "Data-Driven Pronunciation Modelling for Non-Native Speakers Using Association Strength Between Phones", *Proc. ISCA ITRW ASR2000, Paris,* pp. 85-90, 2000.

[7] I.A. Amdal, F.E. Korkmazskiy, A.C. Surendran, "Joint Pronunciation Modeling of Non-Native Speakers Using Data-Driven Methods", *Proc. of ICSLP2000, Beijing,* pp. 622-625, 2000.