Dynamic Lexicon Using Phonetic Features

Kyung-Tak Lee, Christian J. Wellekens

Institut Eurécom, Sophia Antipolis, France
{lee,welleken}@eurecom.fr

Abstract

In order to better model pronunciation variations, we present in this paper a method to build a lexicon whose content changes dynamically with the input speech. To achieve this goal, we proceeded in two steps. In the first step, a static augmented lexicon is created by adding new phone transcriptions to a basic lexicon. These new variants are derived from phonetic features that are automatically extracted from some training speech. Then in the second step, phonetic features are extracted again during recognition and help to select entries in the augmented lexicon that best match the phonetic characteristics of a given speech. These selected transcriptions constitute the dynamic lexicon, which is specific to each input utterance. Experiments showed a 16.0% relative reduction in WER compared to the baseline and 16.7% compared to when a static augmented lexicon is used.

1. Introduction

Even though standard ASR systems are capable of handling variabilities of speech to a certain extent, for example by using Gaussian mixtures in an HMM system, their performances are limited because they generally admit only one possible pronunciation per word. Such simplification does not represent well the reality, as speech characteristics depend on a great deal of inter- and intra-speaker factors and contexts that influence pronunciations. Consequently, use of a single phone transcription per word can not only limit performance in recognition, but also lead to an incorrect (re)estimation of acoustical models in training if the baseform transcription is significatively different from the one actually uttered.

Many previous experiments showed the evidence that ASR performance can be improved by explicitly taking account of pronunciation variations. As a survey of the literature on this topic shows ([1]), pronunciation modeling can be applied to different parts of a system. The most common way is to work at the lexicon level, by simply adding new transcriptions to a basic lexicon. However, it is also known that adding too many variants increases confusability between them and hence limits and sometimes even decreases the recognition rate.

The purpose of our work is to try to limit this confusability by selecting dynamically only the most relevant transcriptions for a given speech. The method will be explained in this paper which is organized as follows. Section 2 describes the different steps to first build a static augmented lexicon. Section 3 explains how the static lexicon can be made dynamic during recognition. Section 4 illustrates the experiments carried out to put these methods into practice, and is finally followed by a conclusion in section 5.

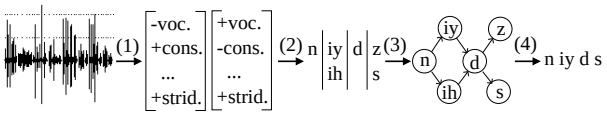

Figure 1: *Steps to build a static augmented lexicon*

2. Static augmented lexicon building

2.1. Overview

The objective of this first part is to automatically discover possible relevant transcriptions for each word, in order to generate a lexicon with new variants. A first step consists of generating all possible transcriptions given an utterance, followed by a selection step that chooses the most likely variants among the ones proposed. Generation and selection of new transcriptions are obtained by respecting the following procedure for each training utterance (Figure 1) :

1. Some phonetic features are first extracted from the input speech on a frame-by-frame basis. A N-best paradigm was adopted to search for the N-best combinations of features per frame.
2. Each combination of detected features for a given frame is mapped to a phone using a lookup table.
3. Successive frames mapped to a same phone are grouped to form hypotheses, which are then connected to each other to build a pronunciation network.
4. All possible transcriptions are generated from the network and the most likely ones are selected by means of a two-pass forced recognition and a pruning process.

The static augmented lexicon is obtained by adding all the selected transcriptions to the basic lexicon.

2.2. Generation of variants

2.2.1. From speech to phonetic features

A classical way to generate new variants is to use a phone recognizer : a string of phones is first recognized from speech, then this sequence is compared to its corresponding lexical transcription through a dynamic programming to obtain variants of pronunciation. In this work, we preferred to base our recognition on phonetic features (listed in Table 1), which are more elementary constituents than phones and are based on linguistic knowledge. Some motivations for this choice are :

- Several linguistic papers (e.g. [2]) mention that use of phonetic features provides a simple framework to understand and capture pronunciation phenomena in speech.

- A recent experiment [3] shows that accurate classification of phonetic features is important for better recognition performance in spontaneous speech.
- Use of phonetic features provides a simple way to detect *transitional frames* (explained in section 2.2.2).

Among the different existing feature types, we chose the SPE system (Sound Pattern of English [4]) due to its popularity and its ease of representation since its feature values are binary. A single neural network was trained to map a set of acoustic parameters presented at the inputs to a bundle of phonetic features (one output node per feature). The mapping is done on a frame-by-frame basis. The network performs a N-to-M classification. Several output nodes may be activated simultaneously as features are considered as independent.

Since feature outputs are not fully reliable, a N-best paradigm was adopted to obtain the N-best combinations of features per frame instead of a single one. To achieve this, two thresholds were fixed at neuron outputs, T_{min} and T_{max} in addition to the activation threshold T ($T_{min} < T < T_{max}$). Any feature value inside these two thresholds is considered as unreliable and its corresponding feature may be toggled. For example, if the neuron output for the feature "vocalic" is activated (value $> T$), but its value is between T_{min} and T_{max}, two groups of features are then generated, the first with "vocalic" activated, and the second with this feature deactivated, while the other features are kept unchanged for both cases. If more than one feature is estimated as unreliable, the number of necessary groups are generated accordingly by considering all toggling combinations, starting with features with values closest to the activation threshold T. The number of generated groups per frame may vary, since the number of unreliable features is not constant.

2.2.2. From phonetic features to phones

For compatibility with lexicons used in standard HMMs, each group of phonetic features is directly mapped to a phone using a lookup table. By doing this for all combinations of all frames, each frame is associated with one or more phones. However, features change rather asynchronously at phone boundaries ([5]) : some features may take values of the next phone while others may still keep values of the previous phone. Consequently, some groups of features cannot be mapped to a valid phone. Frames for which this case occurs have been called here *transitional frames*. Moreover, toggling unreliable features during the generation of the N-best groups of features per frame may lead to the same behaviour. A frame may therefore correspond to a *transitional phone* (equivalent to a garbage phone) if one or more of its groups of features cannot be mapped to any valid phone (only the "best" transitional phone per frame is considered). Transitional phones and frames were used to score pronunciations during the construction of dynamic lexicons.

2.2.3. From phones to pronunciation network

Successive frames mapped to a same valid phone are grouped together to form hypotheses. There must be at least F_{min} successive frames for a hypothesis to be valid, otherwise it is rejected. Moreover, a hypothesis is considered as reliable if at least R_{min} ($R_{min} \leq F_{min}$) of its frames are associated with only one valid phone, the one represented by the hypothesis. As a frame may be associated with several phones, hypotheses may overlap partially or even totally in time.

Next, hypotheses are linked to each other under some constraints to form a pronunciation network. Constraints check whether two hypotheses are not too far away to be connected, and then test for possible succession and/or substitution between them based on how much they overlap. Once the pronunciation network is built, each single path through this network represents a possible transcription. For the generation of variants per word, the whole network is finally segmented into subnetworks (one per word), according to time boundaries given either by a hand-labelling or, if not available, a forced alignment process using a trained HMM word recognizer.

2.3. Selection of variants

By scanning all possible paths through each pronunciation subnetwork, we can collect new phone transcriptions for each word. All variants per word are then compared against its corresponding canonical transcription by using a first pass of forced recognition using a standard Viterbi algorithm. The best path returned according to the maximum likelihood tells which variants best match the input signal. Any variant preferred to its canonical transcription is added to the basic lexicon to form a first static augmented lexicon.

To further restrict the number of variants and to keep only the most representative ones, a second pass of forced recognition is processed. All transcriptions accepted during the first pass are candidates for the second pass. Since a word may be uttered several times in the database, still many variants per word may be available. So during the second pass, the best phone transcription for a word utterance in the first pass may be preferred to the best transcription for another utterance of the same word. Consequently, the number of variants will be reduced. Finally, transcriptions of the most frequent words (typically function words such as "and") are subject to some pruning depending on their frequencies of occurrence. Namely, variants whose probability of occurrence is lower than a minimum value P_{min} are rejected. The final remaining variants with all the canonical transcriptions constitute the final static augmented lexicon.

3. Dynamic lexicon building

3.1. Overview

The purpose in this second part is to check whether a word is likely to be uttered in an unknown speech, and if so to keep only the most suitable pronunciation among the ones proposed for this word. All available transcriptions come from the static augmented lexicon. The method to achieve this goal is guided by the following steps for each utterance during recognition :

1. A pronunciation network is built following the method used to generate new variants (cf. section 2).
2. For each word in the static augmented lexicon, the network is scanned to find a match with one of the available transcriptions of the word. If the match is good enough, the best matching transcription according to some criteria is selected and is added to the basic lexicon.

The dynamic lexicon is used instead of the basic and static augmented lexicons in a standard HMM recognition. It is called *dynamic* because its content is adapted to each distinct utterance. The next subsections explain how to decide which is the best suitable transcription for a given word and utterance.

3.2. Pronunciation match search

To limit the time spent to search through the pronunciation network, a four step approach including some pruning procedure was adopted for each lexical phone transcription :

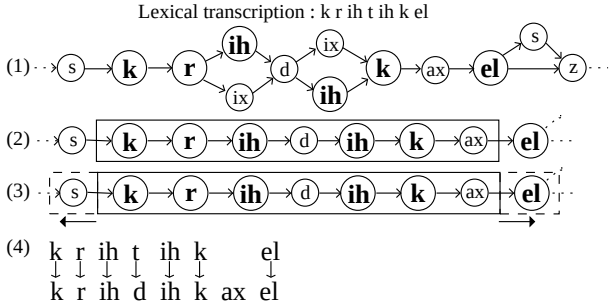

Figure 2: *Steps to search for a match in pronunciation network*

1. Any hypothesis in the network representing a phone found in the lexical transcription is marked.
2. The network is scanned to detect accumulations of marked hypotheses. If no such zone exists, the current lexical transcription is rejected. Otherwise, a start window is placed on an accumulation zone, with a size equal to the number of phones of the lexical transcription.
3. The window is progressively extended on both sides to try to find other distinct marked hypotheses in the area. The process is repeated for all other accumulation zones found. Only windows with the highest number of distinct marked hypotheses are kept, the others are pruned.
4. A detailed match is processed through dynamic programming (DP) between the lexical transcription and each sequence of phones found in the selected windows.

An example is given in Figure 2 for a transcription of the word "critical". A match is considered as valid if it satisfies certain *matching conditions*. If several valid matches are found for the same word, the one with the highest *matching score(s)* is chosen. Conditions and scores are discussed in the next subsection.

3.3. Matching conditions and scores

Pronunciation match search through DP gives the number of substitutions, deletions and insertions occurred with respect to the number of reference phones, hence its matching ratio MR. Conditions for a valid match are : (1) its matching ratio MR must be above a certain threshold MR_{min}, (2) results of DP must not contain any bad mappings, e.g. two successive insertions are forbidden.

Two matching scores were used to select the best transcription. The first one, S_{phn}, is evaluated at phone level. It concerns phones in a lexical transcription that are correctly mapped according to the DP (i.e. no substitutions, deletions or insertions). It is evaluated from the conditional probabilities of appearance of these phones given the word W :

$$S_{phn} = \sum_{i \in \Gamma} \sum_{j=1}^{N_{pron}} P(ph_i|pr_j) \cdot P(pr_j|W) \qquad (1)$$

Γ is the set of all lexical phones that are correctly matched, N_{pron} is the number of distinct pronunciations for the word W, ph_i is the i-th lexical phone correctly matched and pr_j is the j-th pronunciation for the word W. Probabilities are evaluated by counting frequencies of occurrence of phones and pronunciations for each word during the generation of variants.

A second score, S_{feat}, is evaluated at phonetic features level and is used in case two transcriptions have the same phone score S_{phn}. This new score also takes account of all mapping errors. The DP returns associations between two sequences of phones and S_{feat} is the average score over all maps. Let us consider now a single map score S^k_{feat}. In case a phone A is substituted to a phone B (A and B may be identical) over N_{frm} frames, and if each frame is associated with a bundle of N_{feat} features, we can evaluate probabilities of *theoretical* features of phone A to be mapped to *real* features of phone B returned by the system. Assuming frames and features independent, and after taking the logarithm and normalizing, we obtain the following expression for S^k_{feat} :

$$S^k_{feat} = \frac{1}{N_{frm} N_{feat}} \sum_{i=1}^{N_{frm}} \sum_{j=1}^{N_{feat}} \log P(A^{fe_j} \rightarrow B^{fe_j}_{fr_i}) \quad (2)$$

where N_{frm} is the number of frames where the substitution occurs, N_{feat} is the number of available features and $P(A^{fe_j} \rightarrow B^{fe_j}_{fr_i})$ is the probability of the j-th theoretical feature of phone A to be mapped to the j-th real feature of phone B at frame i. Since a neural network is used to output real feature values frame by frame, and all output neuron values are in the range [0,1], each term of (2) is approximated using the following equation :

$$P(A^{fe_j} \rightarrow B^{fe_j}_{fr_i}) \approx 1 - \left| targ(A^{fe_j}) - act(B^{fe_j}_{fr_i}) \right| \qquad (3)$$

where $targ(A^{fe_j})$ is the target value of the j-th feature of phone A (1 if the feature is "on", 0 if it is "off"), and $act(B^{fe_j}_{fr_i})$ is the activation value returned by the j-th output neuron for phone B at frame i.

Probability of a phone B to be inserted between two phones is assumed approximatively equal to the probability of a theoretical *transitional phone* (mentioned in section 2.2.2) T to be substituted by the phone B :

$$P(\emptyset \rightarrow B^{fe_j}_{fr_i}) \approx 1 - \left| targ(T^{fe_j}) - act(B^{fe_j}_{fr_i}) \right| \qquad (4)$$

The idea behind this is to see how much real feature values of phone B are distinct from typical feature values of a transitional phone. If comparisons show little differences, B can be assimilated to a transitional phone that could optionally be present between any valid phones in lexical transcriptions. The insertion of B is therefore plausible and probability of insertion is high. On the contrary, if B is significatively different from a transitional phone, it is likely a valid phone and the fact that B is not present in the lexical transcription must be penalized, hence a lower insertion probability leading to a lower match score. Theoretical feature values of a transitional phone T between two valid phones A_p and A_n are assumed to be :

$$targ(T^{fe_j}) = \begin{cases} 0 & \text{if } targ(A_p^{fe_j}) = targ(A_n^{fe_j}) = 0 \\ 1 & \text{if } targ(A_p^{fe_j}) = targ(A_n^{fe_j}) = 1 \\ 0.5 & \text{otherwise} \end{cases} \qquad (5)$$

Similarly, probability of a lexical phone A to be deleted is assumed approximatively equal to the probability of A to be substituted by a transitional phone T standing between phones B_p and B_n :

$$P(A^{fe_j} \rightarrow \emptyset) \approx 1 - \left| targ(A^{fe_j}) - act(T^{fe_j}_{fr_i}) \right| \qquad (6)$$

If no transitional frame exists between B_p and B_n, boundary frames of these two valid phones are used instead to evaluate the probability.

4. Experiments

4.1. Database and tools

All experiments were carried out on the TIMIT database [6]. All training sentences except SA files were used, as well as the core test set for evaluation. The HMM system used to build the baseline recognizer and to evaluate the different lexicons is HTK [7]. The neural network used to map from acoustic vectors to phonetic features is the NICO toolkit [8].

4.2. Baseline system

Similar experiments as in [9] were followed to build accurate phone models based on hand transcriptions of TIMIT. Models are right-context biphones trained from 39 MFCC coefficients (12 static + 1 energy, 13 Δ, 13 $\Delta\Delta$). Training included data clustering and mixture splitting (6 mixtures per state). The system achieved a 71.9% phone accuracy, a result comparable with the one reported in [9]. The same models were then used to recognize words this time, and obtained a 26.2% WER. A bigram was generated from all sentences of TIMIT for this purpose.

4.3. Phonetic features recognition results

Similar experiments as in [5] were followed to train the neural network. 3596 sentences were used for training and 100 for cross-validation. Comparisons between recognized features and those derived from the hand phone transcriptions of TIMIT led to the results in Table 1, given in percentage of frames correct on the cross-validation set.

Feature	Correct (%)	Feature	Correct (%)
vocalic	88	round	94
consonantal	91	tense	91
high	89	voice	93
back	88	continuant	94
low	93	nasal	98
anterior	91	strident	97
coronal	90	silence	98
Average	**93**	**All correct**	**54**

Table 1: SPE phonetic features recognition results

The results show that each feature taken separately can be reliably recognized. The "all correct" shows how frequently all features are simultaneously correct for a given frame. We see that in average more than one frame out of two is phonetically well-identified, reminded that feature outputs are independent and so $2^{14} - 1 = 16383$ combinations lead to an error.

4.4. Results with new lexicons

The generated static augmented and dynamic lexicons were used instead of the basic lexicon for recognition. The output neuron thresholds T_{min}, T and T_{max} were fixed to 0.25, 0.5 and 0.75 respectively. The minimum number of successive frames for a valid hypothesis (F_{min}) was set to 3, the minimum number of required frames for a reliable hypothesis (R_{min}) to 2, and pruning probability for frequent words (P_{min}) to 0.05. Results are given in Table 2.

The results show that the static augmented lexicon did not improve and even slightly decreased performance. It seems that increase of confusability between lexicon entries counterbalanced any higher transcription accuracy brought by adding new variants. The dynamic lexicons were built using different values of matching ratio thresholds MR_{min}. The best result obtained is 22.0% WER, so a 16.0% relative reduction compared to the baseline and 16.7% compared to the static augmented lexicon. The corresponding MR_{min} is around 30% for our system. Below that level, variants are too easily accepted in which some of them may be inaccurate. On the other hand, increasing too much MR_{min} results in rejecting too many variants, in which some of them might be relevant.

Lexicon	WER (%)
Baseline	26.2
Static augmented	26.4
Dynamic	22.0

Table 2: Recognition results with static and dynamic lexicons

5. Conclusion

We showed in this paper that adapting a static lexicon to the input utterance by making its content dynamic can help to improve performance. Moreover, it was pointed out that phonetic features could be reliably recognized and could help to select dynamically appropriate variants for a given utterance. All these experiments were carried out using a read speech database, but future experiments will also include application of this method to spontaneous speech.

6. Acknowledgments

The authors would like to thank Dr Simon King for providing the script to train the neural network used in the experiments.

7. References

[1] Strik, H. and Cucchiarini, C., "Modeling Pronunciation Variation for ASR : a Survey of the Literature", Speech Communication, Vol. 29, Nos 2–4, pp. 225–246, 1999.

[2] Stevens, K., "Applying Phonetic Knowledge to Lexical Access", Proc. Eurospeech-95, pp. 3–11, 1995.

[3] Greenberg, S. and Chang, S., "Linguistic Dissection of Switchboard-Corpus Automatic Speech Recognition Systems", ISCA ITRW ASR-2000, pp. 195–202, 2000.

[4] Chomsky, N. and Halle, M., "The Sound Pattern of English", MIT Press, Cambridge, 1968.

[5] King, S. and Taylor, P., "Detection of Phonological Features in Continuous Speech using Neural Networks", Computer Speech and Language, Vol. 14, No 4, pp. 333–353, 2000.

[6] Lamel, L., Kassel, R. and Seneff, S., "Speech Database Development : Design and Analysis of the Acoustic-Phonetic Corpus", DARPA Speech Recognition Workshop, pp. 100–109, 1986.

[7] Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V. and Woodland, P., "The HTK Book, Version 2.2", Cambridge University Engineering Department, 1999.

[8] Ström, N., "The NICO Toolkit for Artificial Neural Networks", http://www.speech.kth.se/NICO, 1996.

[9] Young, S., Woodland, P., "State Clustering in Hidden Markov Model-based Continuous Speech Recognition", Computer Speech and Language, Vol. 8, No 4, pp. 369–383, 1994.