ITRW on Pronunciation Modeling and Lexicon
Adaptation for Spoken Language Technology
(PMLA2002)
Aspen Lodge, Estes Park, Colorado, USA
September 14-15, 2002

PRONUNCIATION VARIATION MODELING FOR ASR : LARGE IMPROVEMENTS ARE POSSIBLE BUT SMALL ONES ARE LIKELY TO ACHIEVE

Qian Yang, Jean-Pierre Martens

ELIS-RUG
Sint-Pietersnieuwstraat 41, B-9000 Gent
qianyang@elis.rug.ac.be

Pieter-Jan Ghesquiere, Dirk Van Compernolle

ESAT-KULeuven
Kasteelpark Arenberg 10, B-3001 Heverlee
pieter-jan.ghesquiere@esat.kuleuven.ac.be

ABSTRACT

In this paper a previously proposed method for the automatic construction of a lexicon with pronunciation variants for ASR is further developed and evaluated. The basic idea is to transform a lexicon of canonical forms by means of rewrite rules that are learned automatically on a training corpus of orthographically transcribed utterances. The method is evaluated on the TIMIT corpus, using a speech recognizer incorporating context-independent HMMs and a bigram language model. It appears that reductions of the word error rate of up to 35 % are possible to achieve. However, it also appears that it is more likely to obtain much lower gains.

1. INTRODUCTION

In the late nineties, many research groups have attempted to raise the accuracy of automatic speech recognition (ASR) by means of pronunciation variation modeling (PVM) at the symbolic level. I.e., by introducing alternative word pronunciations (pronunciation variants) in the lexicon and even in the language model of the recognizer (see [11] for a survey). Although nowadays, most ASR lexicons do include multiple (usually hand crafted) pronunciations of frequent words [7], the aim of PVM is to retrieve the important variants in a fully automatic way.

In a special issue of Speech Communication in 1999, two papers [1, 2] reported very significant ASR improvements due to PVM, but these improvements were obtained with recognizers incorporating non state-of-the-art acoustic models. Other papers (e.g. [4]) reported improvements on databases that are not widely available for benchmarking to the international speech community. The few papers reporting tests with state-of-the-art recognizers on widely used databases (e.g. [9]) showed only a marginal gain. Since the Special Issue, the proposed methods were further developed (e.g. [14]) or more systematically evaluated [13, 5]. The evaluations showed that PVM can induce a lot of changes in the recognition output, but the positive and negative effects of these changes are almost in balance. It is our impression that up to date automatic PVM did not live up to its promises, a view which is also supported by distinguished people like Mari Ostendorf [10] and Steve Young [17]. We do believe however that better PVM techniques will finally be capable of providing more significant and stable ASR performance gains.

The rest of the paper is organized as follows. Section 2 briefly outlines the proposed methodology. Section 3 describes the newly developed procedures for learning the pronunciation rules and for generating pronunciation variants by means of these rules. Section 4 discusses the integration of a lexicon with variants in an HMM recognizer, and section 5 reviews the performance gains which were obtained with the proposed methodology.

2. OUR PVM METHODOLOGY

The method we describe here is a further development of the one proposed in [1] and subsequently improved in [14]. It presumes that (i) pronunciation variants can be generated by transforming the transcriptions of a baseline lexicon, (ii) these transformations can be described in terms of stochastic pronunciation rules, and (iii) these rules can be learned automatically on a corpus of orthographically transcribed training utterances.

We assume that the baseline lexicon is constructed on the basis of generally available phonological knowledge sources, such as a phonetic dictionary and/or a grapheme-to-phoneme converter.

The stochastic pronunciation rules which will transform this lexicon are rewrite rules of the form

$$r : L\underline{F}R \rightarrow F' \text{ with } P_r$$

The symbols F, L, R and F' represent variable length phoneme sequences, henceforth called patterns. Rule r says that if a focus pattern F, surrounded by left and right context patterns L and R, is found in the baseline transcription of a word, two pronunciation variants can be produced: one with F being transformed to F' (probability P_r) and one with F being left unaltered (probability $1 - P_r$). The patterns L, R and F together form the *rule condition*, while F and F' constitute the rule input and output. Rules will not be considered independently of each other, but as elements of a rule network. Thanks to the stochastic nature of the rules, probabilities can be introduced in the pronunciation variants. These probabilities are henceforth called *pronunciation probabilities*.

To learn the rules, we need two phonetic transcriptions of each training utterance: a *baseline transcription* emerging from the orthography and the baseline lexicon, and a *'correct' transcription* representing the ground truth. The word *correct* must be put between quotes because there is actually no means of knowing unequivocally whether a transcription is correct or not. Moreover, the 'correct' transcription will have to be generated in an automatic way starting from the orthography and a set of sub-word acoustic models. Since the latter are bound to be imperfect, the generated transcription will be imperfect as well. Therefore, we will henceforth use the neutral term *target transcription* when referring to the 'correct' transcription.

The work described in this paper extends our previous efforts [1, 14, 15]. First of all, we have considerably improved the rule

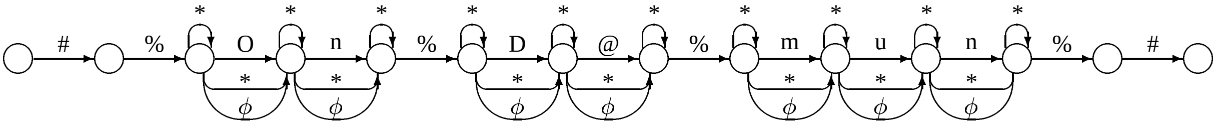

Fig. 1. A graph for producing the target transcription of the sentence *on the moon* with baseline transcription /On D@ mun/ (SAMPA notation). Observe the end-of-sentence phonemes /#/, the word borders /%/ and the substituted and inserted phoneme edges (/*/).

learning process and evaluated our methodology using an HMM-based recognizer (our previous results were all obtained with a segment-based system [12]). The new recognition framework also forced us to conceive a new algorithm for the automatic generation of target transcriptions, and to search for an appropriate way of integrating the pronunciation probabilities into the recognizer.

3. MODELING PRONUNCIATION VARIATIONS

3.1. Target transcription generation

To learn good pronunciation rules, we first of all need a good target transcription of every training utterance. For generating such a transcription we first construct a graph (see Fig. 1) comprising *normal edges* for emitting the phonemes of the baseline transcription, and *error edges* for allowing the generation of alternative transcriptions. Then we convert this graph into an HMM by substituting its phoneme emitting edges by acoustic models. Finally, we retrieve the target transcription from the best alignment of the acoustic vectors with the states of this HMM.

Since PVM is presumed to be more helpful in combination with context-independent (CI) than in combination with context-dependent (CD) acoustic models, and since our first aim is to demonstrate that our PVM approach can improve an HMM recognizer, we assume for the moment that the recognizer incorporates CI-HMMs. Once we have proved our case, we plan to investigate whether an automaton based on CI-HMMs can also generate suitable target transcriptions for the learning of rules that will be used in combination with CD-HMMs.

In the example graph of Fig. 1, we discern three types of error edges: deletion edges (marked with /ϕ/) and substitution and insertion edges (both marked with /*/). Although invisible on the figure, we also attached phoneme independent transition probabilities (P_d, P_s and P_i respectively) to these error edges.

Substituting normal phonemes (including /#/) by their acoustic model is straightforward, but substituting /*/-edges is not. We propose to substitute such an edge by a network consisting of as many parallel branches as there are phonemes in the phoneme inventory (see Fig. 2). We also propose to multiply all the acoustic likelihoods in these branches with an acoustic penalty P_a. By doing so, the aligner will be inclined to substitute/insert a phoneme only when this significantly raises the **average** acoustic likelihood of the frames.

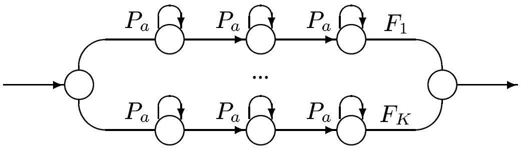

Fig. 2. Acoustic model for an insertion/substitution edge. The phoneme inventory is $\{F_1, .., F_K\}$. An acoustic penalty P_a is assigned to all transitions consuming a frame.

By introducing separate acoustic penalties P_{as} and P_{ai} in substitution and insertion branches respectively, we can easily control the number of substitutions and insertions being generated. The transition probability P_d then controls the number of deletions being generated. The transition probability $P_i < 1$ is needed to favor a single phoneme insertion over the insertion of consecutive phonemes. The transition probability P_s is just used as a boolean that enables or disables the generation of substitutions via substitution edges. However, even if $P_s = 0$, a substitution is still possible: a deletion + insertion tandem will be interpreted as a substitution.

3.2. Rule learning

The rule learning process described in this paper represents a further improvement of the one originally proposed in [1] and further developed in [14]. The newly devised algorithm produces a more compact rule set, and the generated rules are much less affected than before by small changes in the free parameter settings.

As in [14], we first identify in each training utterance the contiguous parts of the baseline transcription which differ from their counterparts in the target transcription. This way we build a list of relevant input/output pairs (F, F'), and, for each pair a list of micro-rules describing the contexts in which the focus is transformed. E.g., if we plan to take two left and two right context phonemes into account in the rule conditions, the micro-rules are of the form /$abFcd \rightarrow F'$/ with $a..d$ representing phonemes.

Once the micro-rules for a particular transformation are retrieved, a hierarchy of more and more general rules for this transformation is induced. In this recursive induction each rule is considered as the child of two parents, obtained by stripping of the leftmost and rightmost context phoneme of the child condition respectively. E.g., /$bFcd \rightarrow F'$/ and /$abFc \rightarrow F'$/ are the parents of /$abFcd \rightarrow F'$/. Figure 3 shows such a hierarchy which is built on three micro-rules with conditions $/abFcd/$, $/cbFce/$ and $/cbFae/$. On each layer, we find rules with the same number of context phonemes in their condition.

In order to assign probabilities to the rules, the baseline-target transcription pairs of the training utterances are parsed from left to right. If F is a possible focus at a particular position - irrespective of whether or not there is a transformation at that position - all networks representing a transformation with input F are examined. The first rule (highest layer, leftmost position) whose condition matches the baseline transcription is selected as the applicable rule for that transformation. This rule is supposed to have fired if the transformation was actually performed. During parsing, we record

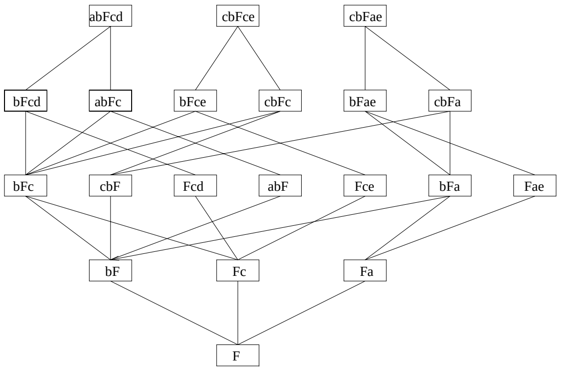

Fig. 3. Hierarchy of pronunciation rules (conditions marked) constructed on the basis of three micro-rules (top level) for the performance of a particular transformation (F, F').

the application frequencies (c_{app}) and the firing frequencies (c_{fir}) of the rules. Since we examine all possible transformations independently of each other at one position before moving to the next position, c_{fir}/c_{app} represents an estimate of the probability of performing the rule transformation, given that the rule is applicable at a certain position.

An important stage of the rule learning process is the pruning of the large rule set that was generated in the previous step. This pruning is described in detail here because it has been changed fundamentally with respect to [14].

A basic operation is the computation of the cost of eliminating a rule. The rule selection mechanism described above implies that if a rule (C) is eliminated, the examples it consumed are taken over by its *first ranked* parent (P). In that case, the frequencies of C must be added to the frequencies of P, thus changing the firing probabilities associated with these examples. In [14] we used the entropy difference ΔH_{CP} and the application frequency c_{app} of C as two separate decision variables. In the present algorithm we accept the elimination of C if the pruning cost

$$\delta_{CP} = \frac{\Delta H_{CP}}{1 + e^{-4\,(c_{app}/c_{th}-1)}}$$

is below some threshold D_{CP}. Obviously, the entropy weighting is intended to favor the elimination of rules with an application frequency much smaller than c_{th}.

Most children happen to have two parents. Therefore, it is not unlikely that interchanging these parents before assessing the pruning cost can lead to a lower pruning cost. In that case, one should prefer this 'child pruning with parent interchange' over 'plain child pruning'. The search algorithm of [14] did acknowledge this, but it ignored the fact that such an interchange can also affect the frequencies of other rules. To suppress the errors due to this effect, the old pruning algorithm had to re-parse the baseline-target transcription pairs every time a layer of rules was processed.

The newly devised algorithm can maintain correct rule frequencies without the need for any re-parsing during the pruning process. The result is that the search now yields a more compact and reliable solution in less time. The basic principles underlying the new algorithm are the following:

- Pruning is performed per network associated with a certain transformation (F, F') and per layer, starting from the top layer.
- Per layer, the pruning is done iteratively. Per iteration, only one rule (the one with the lowest pruning cost) is removed, provided its pruning cost is below a prescribed threshold D_{CP}. This is repeated until no rule is removed anymore.
- When a child is eliminated, its frequencies are added to those of its first ranked parent.
- If adding the child frequencies to the second parent would yield a pruning cost that is smaller than D_{CP} and smaller than the cost for adding them to the first parent, the opportunity of performing pruning with parent interchange is examined. One can show that two neighboring rules on a particular layer can be interchanged without affecting their frequencies, if their conditions are disjunct or if their conjunction is a pattern that is consumed by a rule on a higher layer. Consequently, pruning with parent interchange is only a viable option if the two parents can be interchanged with intermediate rules until they are neighbors (then they can be interchanged as they have a child in common).

Fig. 4 depicts a possible rule network obtained by pruning the network of Fig. 3. Only 8 of the 20 original rules have survived the pruning. Note that in the pruned network, there can be a connection between a rule and its grand parent.

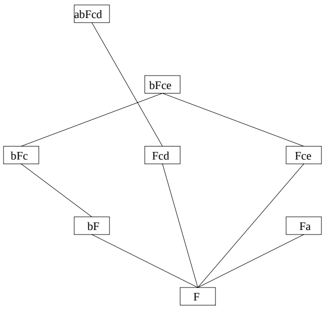

Fig. 4. Pruned network of pronunciation rules (conditions marked) for the performance of a particular transformation (F, F').

3.3. Pronunciation variant generation

Starting with one variant (the baseline transcription of the word), and continuing from left to right, more and more variants can be created by letting rules transform the still untransformed tails of already existing variants. The process is described in detail in [14]. The novelties in the present implementation are the following:

- Due to the new interpretation of the rule probabilities, the probability of not producing any variant is simply one minus the sum of the probabilities of the applicable rules at the given position.
- The decision on whether or not to generate a variant at some position is no longer based on the global variant probability, but on the firing probability of the rule that will be applied for this generation (generate if $P_r > P_{min}$).

Since the rule learning procedure also supports the learning of cross-word rules - defined as rules with a context extending over a word boundary - the variant generation process can be more complicated than suggested here. In fact, in that case the pronunciation variants of a word (and their probabilities) can depend on the identities of the preceding/succeeding words. The variant generation process must then be performed on extended word transcriptions (including word borders and word contexts), and repeated for every extended transcription to consider given the set of cross-word rules. For details on this issue, we refer to [14].

3.4. Final rule pruning

Since variants are only generated by rules with a $P_r > P_{min}$, a final pruning of the rule set is possible. Starting from the bottom layer one can remove all parent-less rules with a firing probability below some P_{th}, equal to the minimal P_{min} one anticipates to need during variant generation. The process can be repeated until no such rules exist anymore. Note that low probability rules with parents cannot be removed because they can act as negative rules inhibiting the firing of more general rules (see [1]).

4. INTEGRATION IN AN HMM RECOGNIZER

4.1. Interconnecting word pronunciation models

All pronunciation variants of a word can be collected in one word pronunciation model, represented as a labeled box with labeled input and output ports (see Fig. 5). The box label is the baseline transcription of the word, the port labels indicate the conditions that must be met by words connecting to this port. If there are no cross-word rules, the models reduce to single input single output models with unconditional (unlabeled) input/output ports.

We always allow two connections from word W_i to word W_j: a direct connection (via a short-pause model *sp*) and a garbage connection (via a garbage model *gm*) (see Fig. 5). The direct connection starts at the W_i output whose label best matches the head of the W_j box label (count number of matching phonemes), and ends at the W_j input whose label best matches the tail of the W_i box label. The garbage connection runs between the unlabeled ports of the two words. Obviously, in the absence of cross-word rules, the two connections always run between the same ports.

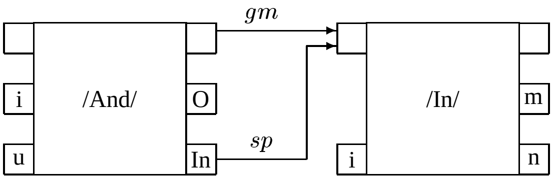

Fig. 5. Connecting the pronunciation models of the words *and* and *in* whose baseline transcriptions are specified as box labels.

4.2. Integrating pronunciation probabilities

It is well known that transition probabilities do not significantly affect the performance of an HMM recognizer. Consequently, just introducing the pronunciation probabilities in a straightforward manner (we call this strategy PVM0) is not expected to make these probabilities very effective. Nevertheless, our experience with PVM and segment-based recognition is that a large part of the PVM gain stems from the pronunciation probabilities [15]. We therefore tried to force the pronunciation probabilities to have more impact on the recognition results.

We have investigated two strategies for increasing the importance of the pronunciation probabilities. In strategy PVM1, we simply raise them to a power $v \geq 1$. In strategy PVM2 we multiply every acoustic likelihood with the corresponding pronunciation probability raised to a power v.

5. EXPERIMENTAL RESULTS

5.1. Speech corpus

All experiments are carried out on TIMIT [6]. This corpus was chosen because (i) it is widely spread in the speech community, (ii) it comes with manual transcriptions against which we can evaluate our target transcriptions, (iii) it is phonetically rich and bound to comprise a variety of pronunciation phenomena, (iv) it is small enough to enable a fast experimentation, and (v) it implies a sufficiently challenging recognition task for which the ASR will produce enough word errors to enable the assessment of relatively small performance gains. Choosing TIMIT also enables us to compare our present results with previously reported ones [1, 14, 15].

Acoustically speaking, TIMIT is read speech of 630 different speakers. It is divided into a training set (462 speakers) and a test set (remaining 168 speakers), and the utterances of 24 test set speakers form the so-called core test set [6].

Linguistically speaking TIMIT comprises 450 SX sentences (vocabulary of 1793 words) and 1890 SI sentences (vocabulary of 5143 words). Each speaker spoke five SX and three SI sentences.

5.2. Experimental conditions

Since we anticipate that phonological knowledge normally offers broad phonetic transcriptions, the pronunciations in the supplied lexicon were converted to US English SAMPA transcriptions (see http://www.phon.ucl.uk/sampa/american.htm). We trained acoustic models for 40 SAMPA phonemes, short pause and silence (garbage). The acoustic models were trained on the full training set, the pronunciation rules on its SX part only.

The acoustic models, the word penalty and the language model power were optimized for the baseline recognizer using HTK [16]. Recognition results were obtained with our own decoder. For the baseline system it produced the same results as HTK. All ASR performances were measured on the core test set: 120 SX utterances (900 words) and 72 SI utterances (665 words).

Tests were performed with two bigram language models. The first one (called BIGTIM1) is the same as in [1]. It allows only word pairs encountered in the 630 x 8 utterances, and it has a perplexity of 9.2. The second language model (called BIGTIM2) is the back-off bigram model that was also used in [8]. It models the TIMIT sentence set (450 SX and 1890 SI sentences) and has a perplexity of 89.2. It is implemented using a back-off node in the search network [16].

By keeping track of the word model output from which the back-off node is reached, and by determining on-line the word model inputs that can be connected to this output, we have realized an efficient but sub-optimal decoder supporting PVM.

5.3. Creating target transcriptions

We consider the target transcriptions appropriate if they meet the following criteria: (i) they are sufficiently close to the true transcriptions, and (ii) they are sufficiently different from the baseline transcriptions.

Since there are indications (e.g. [3]) that substitution errors can be handled to a large extend by the acoustic models, we have generated two target transcriptions per utterance: one obtained by inhibiting the substitution edges (NOSUB) and one obtained by activating these edges (SUB).

Table 1 shows the number of deviations between the target transcriptions and the baseline transcriptions, and between the target transcriptions and the manual transcriptions (derived from the manual labels supplied with the TIMIT data). The latter are presumed to be good estimates of the true transcriptions. The NOSUB transcriptions were obtained with $P_d = 10^{-8}$, $P_i = 0.1$ and $P_{ai} = 0.002$. Keeping the same P_d, P_i and P_{ai}, but activating the substitution edges (with $P_{as} = 0.01$) yields significantly more substitutions. About half of the substitutions between the target and the manual transcriptions seem to be confusions between acoustically proximate vowels (e.g. /@/ versus /I/). Clearly, the NOSUB target transcriptions better meet criterion (i) while the SUB transcriptions better meet criterion (ii).

network	reference	del	ins	sub	total
NOSUB	manual	2.8	2.9	9.1	14.8
	baseline	3.8	1.3	0.1	5.2
SUB	manual	2.4	3.1	10.7	16.2
	baseline	2.9	1.0	4.1	8.0

Table 1. Deviations (in %) between the target and the manual transcriptions and between the target and the baseline transcriptions for the cases NOSUB (substitution edges inhibited) and SUB (substitution edges activated).

5.4. Reference experiment

In Table 2 we have listed the word error rates (WERs) of different systems: (i) two baseline HMM systems: one with 42 CI-HMMs (32 mixtures per state) and one with 6074 cross-word triphone CD-HMMs (6 mixtures per state), (ii) two segment-based systems (denoted as DSSM): one with and one without PVM (data from [14]), and (iii) some CI-HMM systems with PVM (rules were learned on NOSUB targets using $D_{CP} = 0.025$ and $c_{th} = 20$). A distinction is made between strategies for introducing the pronunciation probabilities (PVMx), and between the old and the new rule learning method. The average number of variants/word was controlled by P_{min}.

The figures show that the *relative* improvements caused by PVM are very significant: about 35 % on SX and 15 % on SI. They compare well to the relative improvements found for DSSM [14], in spite of the fact that the CI-HMM system clearly outperforms the DSSM system. Apparently, an ASR comprising CI-HMMs and PVM can compete with an ASR comprising CD-HMMs. A more detailed comparison of the outputs of the ASRs with and without PVM shows that PVM introduces only a limited number of changes: it corrects some errors without introducing new errors. Using phonological rules, Kessens et al [4] found a lot of changes due to PVM, but the good and the bad changes were almost in balance.

system	SX	SI
CI-HMM	2.22	4.81
CD-HMM	1.56	4.21
DSSM	4.03	14.31
DSSM+PVM0	2.12	11.59
old method: 121 rules for $P_{th} = 0.03$		
CI-HMM+PVM0	2.22	6.32
CI-HMM+PVM1 ($v = 40, P_{min} = 0.06$)	1.67	4.06
CI-HMM+PVM2 ($v = 8, P_{min} = 0.06$)	1.44	4.51
new method: 69 rules for $P_{th} = 0.03$		
CI-HMM+PVM0	2.44	6.02
CI-HMM+PVM1 ($v = 40, P_{min} = 0.06$)	1.78	4.06
CI-HMM+PVM2 ($v = 8, P_{min} = 0.06$)	1.44	4.21

Table 2. Word error rates (in %) for the core test set. Systems with pronunciation variants are marked with +PVM. An additional index refers to the pronunciation model integration strategy.

Obviously, PVM0 does not yield any improvement. On the contrary, it causes a degradation. The strategies PVM1 and PVM2 seem to offer comparable gains given the right values of v. Varying v in the range from 30 to 50 (PVM1) does not severely affect the obtained gains. Neither does varying P_{min} in the range from 0.05 to 0.08.

Although the performance gains obtained with the old and the new rule sets are quite comparable, the new rule set is much smaller (69 instead of 121 rules for $P_{th} = 0.03$) and the performance is less affected by the average number of variants per word.

5.5. Changing the language model

Table 3 summarizes the BIGTIM2 results for two baseline HMM systems and two systems with PVM. The 15 % relative gain in performance measured on SI is similar to the one measured before. This may be due to the fact that the perplexity of the SI utterances only changes from 22 to 28 when changing the language model from BIGTIM1 to BIGTIM2. On SI, an ASR with CI-HMMs and PVM is competitive with a CD-HMM system.

system specification	SX	SI
CI-HMM	6.66	8.07
CD-HMM	3.55	6.73
old method		
CI-HMM+PVM1 ($v = 10, P_{min} = 0.06$)	6.22	7.03
CI-HMM+PVM2 ($v = 2, P_{min} = 0.06$)	6.33	6.88
new method		
CI-HMM+PVM1 ($v = 10, P_{min} = 0.06$)	6.10	7.03
CI-HMM+PVM2 ($v = 2, P_{min} = 0.06$)	6.33	7.03

Table 3. System performances (WER in %) for BIGTIM2.

The 8 % relative gain in performance measured on SX is much smaller than the corresponding one attained with BIGTIM1. This may be due to the fact that the perplexity of the SX utterances also changes dramatically (from 6 to 22) when BIGTIM1 is substituted by BIGTIM2. I.e., there is a large mismatch between the statistics of the training corpus and the ones implied by BIGTIM2. This mismatch is argued to be the main responsible for the loss in performance gain.

A detailed analysis of the changes due to PVM showed that in the case of BIGTIM2 a number of errors were substituted by other errors due to PVM.

5.6. Changing the target transcriptions

In this section we compare the performances obtained with rules learned from the SUB and NOSUB target transcriptions (section 5.3). In both cases, the new rule learning process generated 66 deletion and 3 insertion rules. In the SUB case it also generated 13 substitution rules. Since the two target transcriptions were different, the deletion rules were different as well.

The performance gains due to the SUB rules are very similar to the ones due to the NOSUB rules. This seems to generalize the findings of [3] to the case of context-independent acoustic models. The SUB rules correct about 50 % more errors, but they also introduce new errors.

5.7. Need for cross-word rules

In order to estimate the importance of cross-word rules for PVM, we ran the rule learning procedure in a *word internal rules only* mode. The target transcriptions were the NOSUB transcriptions mentioned before, and the number of rules was 69. The performance gains measured on SX remained the same, but the ones measured on SI became considerably worse. In combination with PVM1, there was almost no improvement at all anymore, neither with BIGTIM1 nor with BIGTIM2. We found something similar [15] when testing PVM in combination with segmental acoustic models. The word internal rules correct about the same number of errors, but they introduce more new errors than the cross-word rules do.

5.8. Retraining the acoustic models

Retraining the acoustic models on the basis of our target transcriptions, and generating new target transcriptions and pronunciation rules on the basis of these models did not cause any further improvement of the recognition accuracy. This is in line with the findings of e.g. [4, 9].

6. CONCLUSION

It is demonstrated that pronunciation variation modeling, if implemented in the right way, can lead to significant improvements in the recognition accuracy. It seems possible to build an ASR comprising context-independent acoustic models and a lexicon with pronunciation variants that can compete with an ASR comprising context-dependent acoustic models. Unfortunately, there are also indications that the obtained performance gains are not always robust against changes in the language model. It is a challenge to understand why, and to establish means of solving this problem.

7. REFERENCES

[1] Cremelie, N. and Martens, J., ”In search for better pronunciation models for speech recognition”, *Speech Communication*, 29, 115-136, 1999.

[2] Holter, T. and Svendsen, T., ”Maximum likelihood modelling of pronunciation variation”, *Speech Communication*, 29, 177-192, 1999.

[3] Jurafsky, D., Ward, W., Jianping, Z., Herold, K., Xiauyang, Y., and Sen, Z., ”What kind of pronunciation variation is hard for triphones to model?”, *Proceedings ICASSP* (Salt Lake City), 577-580, 2001.

[4] Kessens, J., Wester, M. and Strik, H., ”Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation”, *Speech Communication*, 29, 193-207, 1999.

[5] Kessens, J., Cucchiarini, C. and Strik, H., ”A data-driven method for modeling pronunciation variation”, *Speech Communication* (to appear)

[6] Lamel, L., Kassel, R. and Seneff, S., ”Speech database development: design and analysis of the acoustic-phonetic corpus”, *DARPA Speech Recognition Workshop*, 100-109, 1986.

[7] Lamel, L. and Adda, G., ”On designing pronunciation lexicons for large vocabulary continuous speech recognition”, *Proceedings ICSLP*, 6-9, 1996.

[8] Lee, K. and Wellekens, C., ”Dynamic lexicon using phonetic features”, *Proceedings Eurospeech*, 1413-1416, 2001.

[9] Riley, M., Byrne, W., Finke, M., Khudanpur, S., Ljolje, A., McDonough, J., Nock, H., Saraclar, M., Wooters, C. and Zavaliagkos, G., ”Stochastic pronunciation modelling from hand labelled phonetic corpora”, *Speech Communication*, 29, 209-224, 1999.

[10] Shafran, I. and Ostendorf, M., ”Use of higher level linguistic structure in acoustic modeling for speech recognition”, *Proceedings ICASSP* (Istanbul), 1021-1024, 2000.

[11] Strik, H. and Cucchiarini, C., ”Modeling pronunciation variation for ASR: A survey of literature”, *Speech Communication*, 29, 225-246, 1999.

[12] J. Verhasselt, J.P. Martens. ”Context modeling in hybrid segment-based/neural network recognition systems,” *Proceedings ICASSP* (Seattle), 501-504, 1998.

[13] Wester, M., Kessens, J. and Strik, H., ”Pronunciation variation in ASR: Which variation to model?” *Proceedings ICSLP* (Beijing), 488-491, 2000.

[14] Yang, Q. and Martens, J.P., ”Data-driven lexical modeling of pronunciation variations for ASR”, *Proceedings ICSLP-2000* (Bejing), 417-420, 2000.

[15] Yang, Q. and Martens, J.P., ”On the importance of exception and cross-word rules for the data-driven creation of lexica for ASR”, *Proceedings IEEE ProRisc* (Veldhoven), 589-593, 2000.

[16] Young, S., Kershaw, D., Odell, J., Ollasson,D., Valtchev, V. and Woodland, P., *The HTK-book, version 3.0*, Cambridge University Engineering Department, 2000

[17] Young, S., ”Statistical modelling in continuous speech recognition (CSR)”, *Proceedings International Conference on Uncertainty in Artificial Intelligence (UAI)*, (Seattle), 2001.