Internal Use Only（非公開）

TR－SLT－0093

Pronunciation Modeling with HMMs as Statistical Lexica

Rainer Gruhn
Konstantin Markov

2005.03.31

概要

Non-native speakers pronounce words in multiple different ways than native speakers. To model these deviations statistically, we propose discrete word HMMs as statistical lexicon. The models are initialized from some baseline lexicon and trained on the result of phoneme recognition. The models are applied by rescoring an n-best recognition result.

（株）国際電気通信基礎技術研究所
音声言語コミュニケーション研究所
〒619－0288「けいはんな学研都市」光台二丁目２番地２ TEL：0774－95－1301

Advanced Telecommunications Research Institute International
Spoken Language Translation Research Laboratories
2-2-2 Hikaridai "Keihanna Science City" 619-0288,Japan
Telephone:+81-774-95-1301
Fax :+81-774-95-1308

Abstract

Non-native speakers pronounce words in multiple different ways compared to native speakers. To model these deviations statistically, we propose discrete word HMMs as statistical lexicon. The initialization of the HMMS bases on a standard pronunciation dictionary. One HMM is generated per word in the dictionary, with one state per phoneme in the baseline pronunciation. Non-native training data is segmented into word chunks, on which phoneme recognition is performed. The probability distributions of the HMMs are trained on the phoneme sequences.

To apply the models, both an n-best word level recognition and a utterance-level phoneme recognition of the test data are required. A pronunciation score is calculated by performing a Viterbi alignment with the HMM dictionary as model and the phoneme sequence as input data. This score is a measure how well the phonemes match with the pronunciation of the word sequence. The hypothesis with the highest score is selected as recognition result. Experiments performed on the ATR SLT non-native English database resulted in a word error rate improvement from 45.88% to 42.14%.

非母国語話者の発音には母国語話者に比べて種々の差異が見られる．本稿では，この差をモデル化するための手法として，HMMを用いた発音辞書を提案する．各単語に対するHMMは，まず通常辞書における音素列の各音素を各状態とする形で生成される．続いて，実際の非母国語話者の発声データを用い，それに含まれる単語の音素認識結果を用いて出力確率と遷移確率が学習される．

このモデルの使用にあたっては，通常の単語認識結果のN-bestと音素認識結果が必要となる．各N-best単語系列に対し、HMMを用いた発音辞書を使い、音素認識結果の音素系列のビタビ・アライメントを得る。その時のスコアを対応する単語系列の発音スコアとする。最終的に，N-bestの中で最も高いスコアを示すものが，認識結果として選ばれる．当研究所の非母国語英語データベースで行った実験で単語誤り率が45.88%から42.12%に下がった。

Contents

Chapter 1

Introduction

There are several reports in literature about pronunciation modeling in general [SC99] and for the special case of non-native speakers [vC01]. Many approaches follow the similar basic scheme of comparing manually or automatically generated phoneme transcriptions to some baseline transcription. Variation information can be extracted from the differences. Typically it is represented in the form of rules, which can be weighted based on occurence frequency, likelihood, confusability or other measures (e.g. [GMN02]). These rules are applied to a baseline lexicon in order to generate some adapted lexicon or to optimize an acoustic model [Tom00]. Unfortunately this approach usually achieves only limited improvement [BGN02].

Other researches are based on the knowledge-based approach of inserting additional phonemes to the dictionary and acoustic model [UB99]. This multilingual approach assumes that non-native speakers use phonemes from their own language that are similar to the foreign language. Those phonemes can be included as pronunciation variants in the dictionary, resulting in some improvement in recognition accuracy. But rule-based approaches are less flexible than data-driven approaches and as more non-native databases become available (e.g. [MTY$^+$04, CGN04]), automatic modeling of non-native pronunciation is the more promising approach.

In this research, we suggest a new data-driven approach to deal with pronunciation variations. It is based on word-level pronunciation HMMs.

The concept of generating HMMs to model pronunciation has been analyzed earlier for automatically generated acoustic subword units. This method has been applied to an isolated word task with one Norwegian speaker [Pal90] to generate pronunciation dictionaries and for a database of 150 Korean speakers [YO99].

In this research, we focus on continuous speech recognition of non-native speakers. With their high pronunciation variability, they are a very promising target for such a statistical approach. The approach is phoneme-based, making the model capable of handling words that are in the dictionary but unseen in the training data, as baseline pronunciations can be retained. The pronunciation HMMs are applied by calculating a pronunciation score for each hypothesis of an n-best recognition with the Viterbi alignment algorithm.

Similar to the standard approach of extracting pronunciation confusion rules, we generate a phonetic transcription with a phoneme recognizer. These phoneme string sequences are used as training data for discrete word HMMs; one HMM for each word. There is no attempt to explicitly represent the phoneme variations. Even phoneme substituions unseen in the training data are allowed, as a certain floor probability exists for all possible phoneme sequences for each word. Insertions and deletions are also modeled implicitly. The HMM training process takes care of all variation- and likelihood issues, unlike in other approaches. E.g. rule firing frequencies, thresholds to determine whether a rule is applicable or not, do not have to be calculated.

Chapter 2

Word HMMs

2.1 Generation

As illustrated in Figure 2.1, two levels of HMM-based recognition are involved in this approach:

- Acoustic level: phoneme recognition to generate the phoneme sequence S_i from the acoustic features O_i
- Phoneme label level: For training, the phoneme sequences S_i are considered as input. For all words, a discrete word HMM is trained on all instances of that word in the training data. The models are applied for rescoring, generating a pronunciation score given the observed phoneme sequence S_i and the word sequence.

The first step requires a standard HMM acoustic model, and preferably some phoneme bigram language model as phonotactic constraint. The continuous training speech data is segmented to word chunks based on time information generated by Viterbi alignment. Acoustic feature vectors are decoded to an 1-best sequence of phonemes.

For each word in the vocabulary, one discrete untied HMM is generated. Figure 2.2 shows as an example the HMM for the word "and".

The models are initialized on the phoneme sequence in some baseline pronunciation lexicon. The number of states for a word model is set to be the number of phonemes in the baseline pronunciation, plus enter and exit states. Each state has a discrete probability distribution of all phonemes.

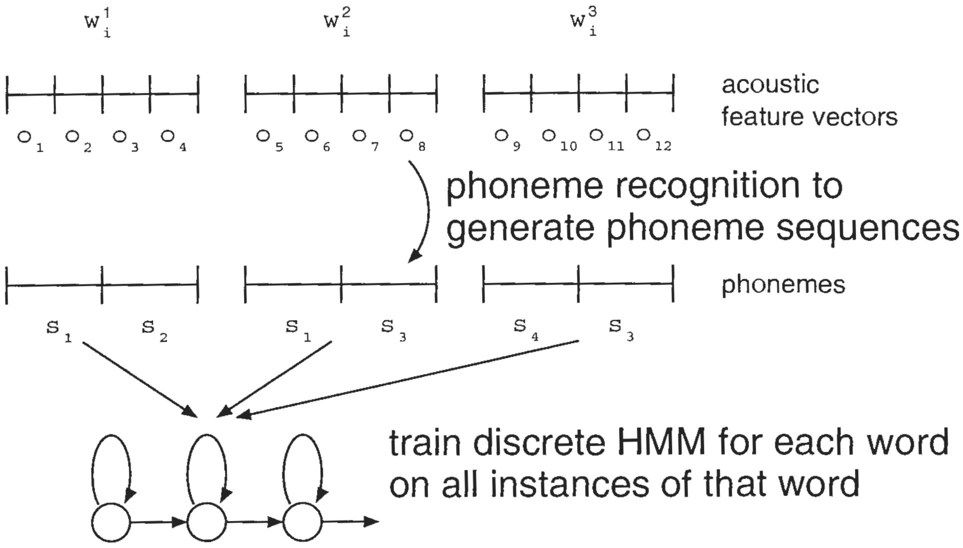
w_i^1
w_i^2
w_i^3
acoustic feature vectors
o_1 o_2 o_3 o_4 o_5 o_6 o_7 o_8 o_9 o_{10} o_{11} o_{12}
phoneme recognition to generate phoneme sequences
phonemes
s_1 s_2 s_1 s_3 s_4 s_3
train discrete HMM for each word on all instances of that word

Figure 2.1: *Two layers of processing are required to generate pronunciation models: an acoustic level for phoneme recognition and the phoneme label level for word model training.*

The phoneme sequence(s) in the baseline dictionary are given a high probability and all other phonemes some low but non-zero value. Forward transition between all states is allowed, with initial transition probabilities favouring a path that hits each state once.

2.2 Training

The probability distribution as well as the transition probabilities are reestimated on the phoneme sequences of the training data. For each word, all instances in the training data are collected and analyzed. The number of states of each word model remains static. Phoneme deletions are covered by state skip transitions, phoneme insertions are modeled by state self-loop transitions.

Data sparseness is a common problem for automatically trained pronunciation modeling algorithms. In this approach, pronunciations for words that do appear sufficiently frequent in the training data, the pronunciations are generated in a data-driven manner. For rare words, the algorithm falls back on baseline phoneme sequences from a given lexicon. This combination should make it more robust than for example an application of phoneme confusion rules on a lexicon (as e.g. in [GMN02]) could be.

2.3 Application

As Figure 2.3 shows, the pronunciation word models are applied by rescoring an n-best recognition result. On a non-native test utterance, both a 1-best phoneme recognition and a n-best (word-level) recognition step are performed.

In standard Viterbi alignment, a speech signal is aligned to a reference text transcription using an acoustic model, with an acoustic score as a by-product. In this approach, the time-aligned lattice is of no interest, although usually it is the main target of Viterbi alignment. Figure 2.4 gives a graphical explanation.

With the pronunciation HMMs as "acoustic model" and each n-best hypothesis as reference, a Viterbi alignment results in an "acoustic score", which is in fact the pronunciation score. Together with the language model score of that n-best hypothesis, a total score is calculated.

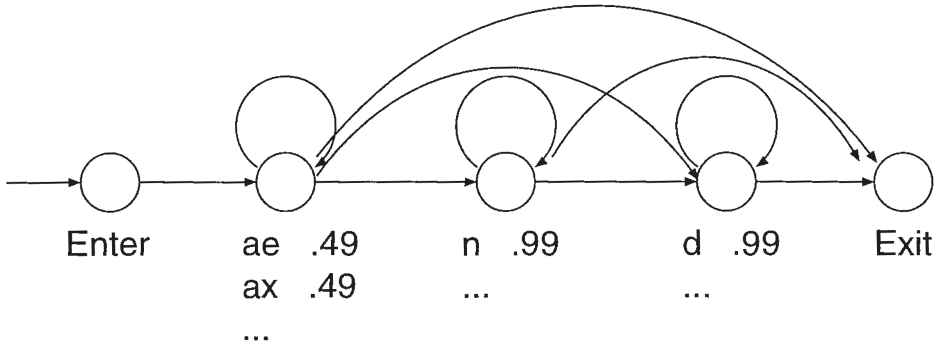

Figure 2.2: *An example discrete word HMM for the word "and", initialized with two pronunciation variations for the first phoneme.*

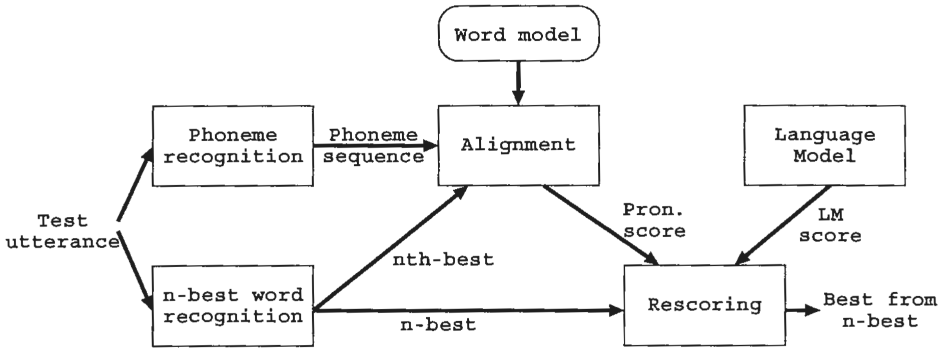

Figure 2.3: *Rescoring an n-best recognition result with word pronunciation models.*

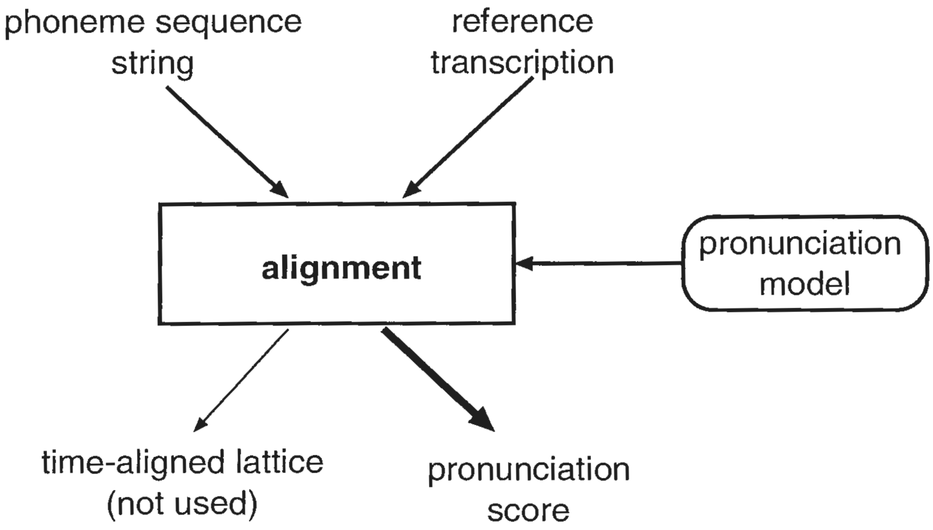

Figure 2.4: *The Viterbi alignment algorithm is used to calculate the pronunciation score.*

Chapter 3

Experiments

3.1 Non-native database

The non-native database was collected at ATR and consists of 90 speakers of English. The first languages of the speakers are Chinese (mostly Mandarin) (CN), French (FR), German (GER), Indonesian (IN) and Japanese (JP). About 14 minutes of read speech are available per speaker. The sentences include six hotel reservation dialogs, TIMIT phonetically balanced sentences and credit-card style digit sequences. The text is uniform for all speakers. Two of the hotel reservation dialogs were chosen as a test set of about three minutes, the rest of about eleven minutes as training data. The number of speakers is shown in Table 3.1.

Table 3.1: *Number of speakers per nation.*

	CH	FR	GER	IN	JP
# speakers	17	15	15	15	28

Some experiments focus on a development set, which is a subset consisting of 11 Japanese speakers.

3.2 Word HMM initialization

The discrete probability distribution for each state is initialized depending on the "correct" phoneme sequence(s) as given in the lexicon. The correct phoneme has a probability of 0.99. If more than one pronunciation variant is included in the lexicon, the variations all have the same probability, totalling 0.99. All other phonemes are assigned some non-zero probability.

The transition probabilities depend on the number of succeeding phonemes in the baseline lexicon. The probability to skip k phonemes is initialized to 0.05^k. Insertions are allowed with a chance of 0.05. The transition to the next state therefore has a probability of slightly below 0.9.

3.3 Phoneme recognition

As a data-driven approach, the pronunciation modeling method proposed here includes a phoneme recognition step. For native speakers, context-dependent acoustic models achieve higher accuracy than monophone models. To examine the impact of context for non-native speakers, phoneme recognition was performed on full utterances with a monophone, right-context biphone and triphone model. All models are trained on more than 60 hours of native English speech data from the LDC Wall Street Journal (WSJ) read newspaper speech corpus [PJ92]. The phoneme set consists of 43 phonemes plus silence. The three acoustic models have the following properties:

- the monophone HMM model has 132 states and 16 mixtures,
- the biphone model 3000 states and 10 mixtures,
- the triphone model 9600 states and 12 mixtures.

The word error rates of these models for the (native English) Hub2 5k task are 19.2%, 15.2% and 6.4%, respectively. The features are 12 MFCC coefficients, energy and the first and second level derivatives.

Table 3.2 shows the phoneme accuracy for monophone, biphone and triphone models on the non-native data. A phoneme bigram model trained on the result of a forced alignment of native speech (WSJ) provided some phonotactic constraint. The references for evaluation are generated automatically from a baseline lexicon. If a correct phoneme transcription was available, higher numbers could be expected. The monophone model performs best for all speaker groups. Obviously, the phonetic context for native English speakers is considerably different to non-native speakers.

Table 3.2: *Phoneme accuracy in %, compared to a canonic transcription.*

	CH	FR	GER	IN	JP
monophone	39.21	45.41	48.85	43.31	37.74
biphone	29.54	37.87	41.15	33.84	29.24
triphone	30.07	41.57	45.45	27.08	29.46

For the rescoring step, the phoneme sequence of the whole utterance is recognized. For the training of the word models, the non-native training data set is segmented into single words based on time information aquired by Viterbi alignment. On these word chunks, phoneme recognition is performed.

The HTK toolkit [WY93] is used for all training and decoding steps.

3.4 N-best word recognition

The HMM pronunciation models are applied in the form of rescoring the n-best decoding result. The n-best recognition uses the monophone acoustic model introduced in Section 3.3 and a bigram language model. Two types of dictionaries have been the base of both pronunciaiton HMM creation and n-best recognition, a LVCSR dictionary with 8875 entries for 7311 words is used in the main experiment. Some experiments that focus on a development set consisting of a group of Japanese speakers of English were conducted with a task-specialized hotel reservation topic dictionary of 6650 entries for 2712 words.

We chose to examine 10-best recognition in this research.

3.5 Rescoring

On each utterance in the test data, both a 1-best phoneme recognition and a standard n-best recognition (on word level) is performed. For each of the n-best word sequences, we apply a forced alignment using the discrete pronunciation models, the phoneme sequence as input features and the word sequence as labels. The resulting score is the pronunciation score.

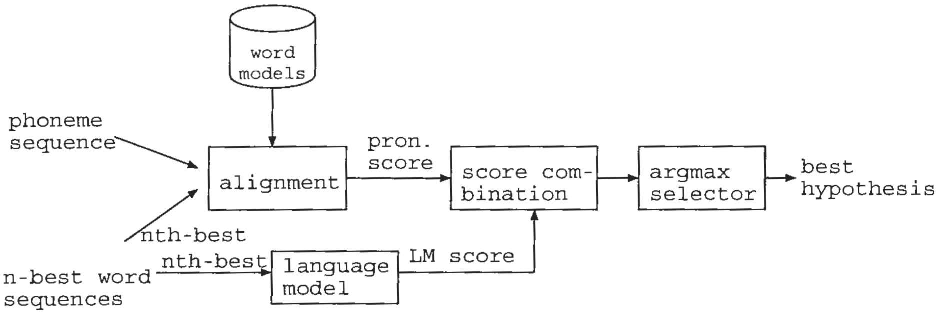

Figure 3.1: *The rescoring process.*

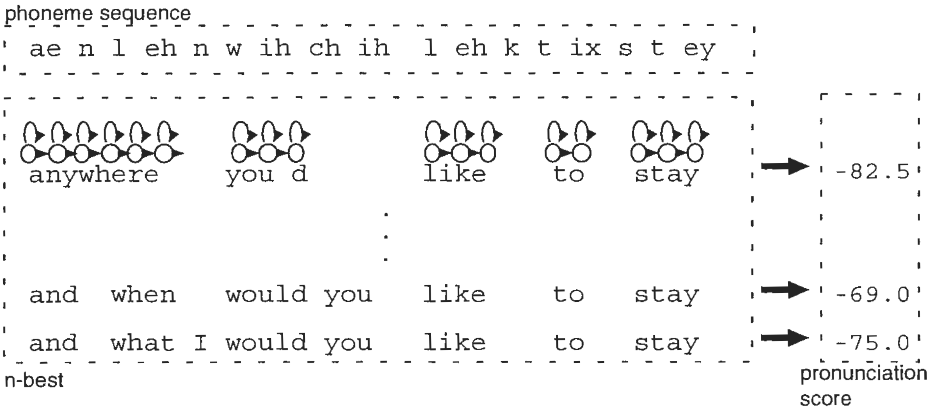
phoneme sequence
ae n l eh n w ih ch ih l eh k t ix s t ey
anywhere you d like to stay
-82.5
and when would you like to stay
-69.0
and what I would you like to stay
-75.0
n-best
pronunciation score

Figure 3.2: *For each n-best hypothesis of an utterance (bottom three lines), a pronunciation score is calulated relative to the phoneme sequence (top line). The correct result is "and when would you like to stay".*

Figure 3.2 shows an example of calculating the pronunciation score for three recognition hypotheses of the utterance "and when would you like to stay". On the phoneme sequence in the top line, an alignment is performed with each hypothesis as transcription. The score is highest for the correct word sequence. Because of mispronunciation and phoneme recognition errors, the phoneme sequence is only similar to the baseline pronunciations of the words.

This pronunciation score is combined with the weighted language model score for this hypothesis. The hypothesis achieving the highest total score among the n-best is selected as correct.

Table 3.3: *Word error rates in % for non-native speech recognition without and with pronunciation rescoring.*

	CH	FR	GER	IN	JP	avg
baseline	51.23	37.93	31.77	40.48	56.92	45.88
rescoring	45.12	34.80	29.88	38.31	52.36	42.14

Table 3.3 shows the word error rates for recognition of non-native speech of the five speaker groups. The larger LVCSR dictionary was used in this experiment. For all speaker groups, the recognition performance could be improved by rescoring the n-best. Averaging over all language groups while considering the number of speakers in each group, the word error rate dropped from 45.88% to 42.14%. Both the highest absolute gain (6.11%) as well as the best relative improvement (11.93%) was archieved for the Chinese speakers. As the size of the database is somewhat limited, it is possible that the Chinese speakers in this database incidentally have the most similar speaking style and English skill, therefore the modeling is most effective for them. An evaluation of their English skill can help analyzing this effect.

Figure 3.3 shows detailed results obtained on the development set with the smaller dictionary for various language model score weights. The baseline performance of 32.54% word error rate can be improved to 29.04%. The correct choice of the language model score weight is very important, in this experiment a factor of 5 was the optimum.

The pronunciation HMMs are initialized from the baseline pronunciation dictionary, then several reestimation iterations modify the probabilities. The effect of this training can be seen in Figure 3.4. Most improvement can be gained with the initial models already, from 32.54% to 29.88% WER. The first training iteration reduces the WER to 29.11%, further iterations bring only minor improvement. Limited coverage of the test data due to small training data may be the reason why the effect of increased training is limited.

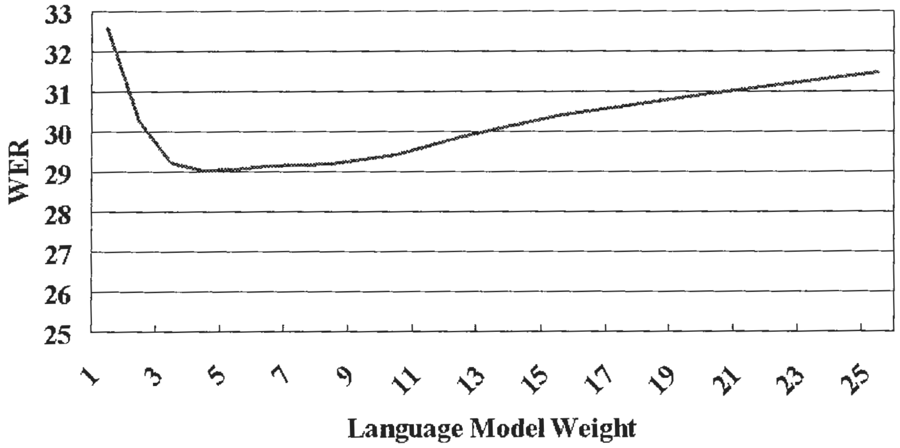

Figure 3.3: *Word error rate for rescoring of n-best based on prounciation score combined with weighted language model scores.*

3.6 Acoustic score

In the previous experiment, the pronunciation score was combined with a weighted language model score. Rescoring only on the basis of the pronunciation score did improve the word error rate. But the pronunciation information alone did not perform as well as when language model information was added.

Another possibility is to take the acoustic score into account as well. The acoustic score for each of the hypothesises is calculated at the n-best recognition step and therefore do not cause any additional computation cost. The acoustic score can be weighted relative to the pronunciation (and language model) scores. But it turnes out that considering the acoustic score for rescoring does not archieve any improvement. The results of an experiment conducted on the smaller set of Japanese speakers is shown in Figure 3.5. The baseline system considers only pronunciation and language model score, the language model weight is set to 5. Independent from the acoustic score weight, the baseline system always performs better.

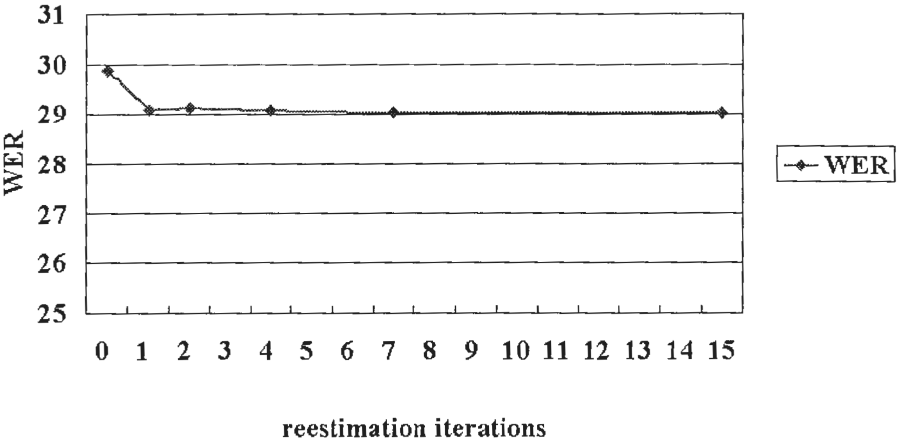

Figure 3.4: *Word error rate for rescoring of n-best based on prounciation score combined with weighted language model scores.*

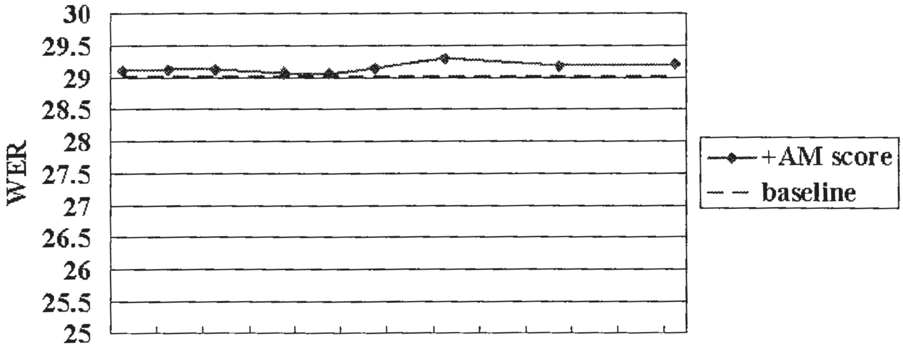

Figure 3.5: *Considering the acoustic score additionally to pronunciation and language model score does not lead to WER reduction.*

Chapter 4

Conclusion

Word error rate could be improved in average from 45.88% to 42.14% with pronunciation rescoring, showing the effectiveness of the approach for non-native speech. The full strength of the approach may not be achieved in this evaluation because the non-native training data covers only a limited share of the total vocabulary. Many word models just default to the standard pronunciations. This will always be a problem in a large vocabulary scenario. It could be countered by extending the training data to other non-native databases, e.g. [MTY$^+$04]. Alternatively, modeling pronunciation on other levels than words may be a solution, but as the English language has a high number of syllables, the coverage problem might worsen in case syllable-level pronunciation is modeled. Considering the acoustic score together with pronunciation and language model score did not improve the performance of rescoring.

Possible future work could include taking the speakers English skill into account by providing skill-dependent pronunciation models. It may also be helpful to initialize the transition probabilities in the pronunciation models based on an examination of typical insertion and deletion error frequencies.

Appendix A

Software

A.1 Overview

This section gives an overview on the typical procedure for training and application of pronunciation HMMs. The steps are explained briefly and include command examples. The software consists of the HTK Version 2.2 toolkit and several programs that are explained in this appendix. The documentation does not include the steps to train a standard HTK format acoustic model and (bigram) language model.

A.1.1 Training

Training data segmentation: The training data needs to be segmented into word chunks. Phoneme recognition accuracy is higher if performed on short segments, and cross-word segments are avoided. The necessary time information is aquired through a Viterbi alignment step with HTK `HVite`. For the physical split of the wave files, the script `DB2words_nphone.py` is provided.

Phoneme recognition: After generating file lists for the language/speaker group to be analyzed, phoneme recognition is performed with HTK `HVite`. Typically, a phoneme bigram provides some constraints.

Feature vector generation: The phoneme recognition result must be converted from the text result format into HTK feature vector format. The script `mlf2phnbin.py` also generates a file list as side output. Users who wish to write their own conversion program are reminded that HTK Headers are always in BigEndian byteorder, even if the data is in LittleEndian byteorder.

Pronunciation HMM initialization: The script `Lex2InitAM.py` reads a standard pronunciation dictionary and outputs a file with one discrete HMM per word in the dictionary.

HMM training: After removing words from the training data, that have not been in the lexicon, with `mlfCheck.py` the training is performed with `HERest` or `LSFherest`. Repeat this step as needed.

A.1.2 Application

The application of word pronunciation models described in this research is by rescoring an n-best list. It is based on both HTK 2.2 and HTK 3.X as well as some python programmes. The procedure is as follows:

Phoneme recognition: On the test data, we perform a phoneme recognition with `HVite`. Accuracy on word chunks would be higher, but as the prerequisite reliable pre-segmentation is not possible, the whole utterance is recognized.

N-best recognition: On the test data, an n-best recognition is performed with `HVite`.

Rescoring: All evaluation steps, such as parsing the n-best and phoneme recognition files, calculating the scores and creating the target file are included in the script `nb_align_rescore.py`.

A.1.3 Usage example

This section gives a step-by-step example on how the experiments described in this report were conducted. There are some incompatibilities between HTK 2 and HTK 3, the all the training and most of the experiments require HTK 2 (and were conducted under HTK 2.2).

The first step is Viterbi Alignment with `HVite` to get the time information for the training data to make a segmentation into word chunks possible. The python script DB2words_nphone.py provides the actual physical information. That script, takes no command line options, the settings are in the initial segment of the script file.

```
cd /data/tetsu4/rgruhn/WSJ/nn/rbiphone_full

find /DB/MDB/EDB1/SPH/WAV -follow -noleaf  | grep 16k | grep
-v "A_" | grep -v "B_" | sort > ! allutt.list

HVite -A -T 1 -l "*" -a -C ./config.align.bigram -S allutt.list
-I allutt.mlf -H /data/tetsu6/xtcinca/adapted_mono/basemodels/h
mm.mono.mix16 -b \!ENTER -i allutt.align.mlf -m -y lab /data/te
tsu4/rgruhn/WSJ/LM/ITLdemo/hrt.1000.lex.open99.v3.mod /data/tet
su6/xtcinca/adapted_mono/basemodels/monophones

python /data/tetsu4/rgruhn/WSJ/tools/DB2words_nphone.py
```

Then, word file lists are created with find, separated by languages (in this example: only German natives):

```
cd scp/phoneme_train
touch GER.scp
foreach i ( M001 M006 M007 M008 M010 M021 M034 M036 M051 M052 M0
54 M056 M071 M076 M078 M079 M092 )
find /data/tetsu6/rgruhn/nn/all_monophone/wrd_wav | grep $i | gr
ep 16k | grep -v TAS22001 | grep -v TAS32002 >> GER.scp
end
```

These file lists are used for the phoneme recognition. In the phoneme recognition, a phoneme bigram and a dummy dictionary (basically a list with the same phoneme in two rows) are applied.

```
cd /data/tetsu4/rgruhn/WSJ/nn/rbiphone_full/phonerec

HVite -H /data/tetsu6/xtcinca/adapted_mono/basemodels/hmm.mono.mi
x16 -S ../scp/phoneme_train/GER -l '*' -i GER.s12.mono.mlf -s 12
-w /data/tetsu4/rgruhn/WSJ/LM/bigram_monophone.net -p 0.0 -C ../c
onfig.phonerec.bigram /data/tetsu4/rgruhn/WSJ/LM/mono_nosp_EE.dic
t /data/tetsu4/rgruhn/WSJ/LM/mono_nosp
```

Transfer word-segment phoneme recognition results into discrete feature vectors. The program mlf2phnbin.py takes the language as argument, the other settings are in the initial segment of the script.

```
python /data/tetsu4/rgruhn/WSJ/tools/mlf2phnbin.py GER
```

We create an initial word pronunciation HMM from the a standard pronunciation dictionary.

```
python /data/tetsu4/rgruhn/WSJ/tools/Lex2InitAM.py
```

To make sure only those words are in the training data that are also in the lexicon:

```
python /data/tetsu4/rgruhn/WSJ/tools/mlfCheck.py GER
```

The actual training (repeat steps with increasing HMM index as desired):

```
mkdir /data/tetsu4/rgruhn/WSJ/nn/hmm/ITLdemo/8k/

HERest -C /data/tetsu4/rgruhn/WSJ/HTK/config -S /data/tetsu4/rgruh
n/WSJ/nn/rbiphone_full/scp/wrdphn_train/8k/${i}_noOOV.list -H /dat
a/tetsu4/rgruhn/WSJ/nn/hmm/ITLdemo/8k/JP/hmm0/MODELS -M /data/tets
u4/rgruhn/WSJ/nn/hmm/ITLdemo/8k/${i}/hmm1 -I /data/tetsu4/rgruhn/W
SJ/nn/rbiphone_full/allwords.mlf -m 2 /data/tetsu4/rgruhn/WSJ/nn/h
mm/ITLdemo/8k/wordlist.txt
```

Now that the models are trained, we move to the application steps. For rescoring, typically both pronunciation and language model scores are considered. As the calculation of language model scores requires a program that is only available in HTK 3, the following extra step is necessary. It must be noticed that the LPlex tool requires <s> and not !ENTER type keywords for sentence start and end.

```
cd /data/tetsu4/rgruhn/WSJ/nn/rbiphone_full/

/usr/bin/python ./nb_align_rescore.py 0

/home/pxn014/kmarkov/soft/htk/bin.linux/LPlex -T 11 /data/tetsu4/rg
ruhn/WSJ/LM/ITLdemo/open99.wbigram.ss alignword.mlf > alignword.lms
```

After that, the main experiment can be run automatically for various language model scale settings. The script includes all steps until the creation of the mlf file with the chosen best hypothesis.

```
foreach lms ( 00 01 02 03 05 07 10 12 15 17 20 22 25 27 30 )

python ./nb_align_rescore.py $lms

end
```

Appendix B

Paths

Some important directories

`/DB/MDB/EDB1` : non-native English database

`/DB/MDB/EDB1/INFO` : speaker information, including pronunciation scores

`/DB/MDB/EDB1/SPH` : speech and label files

`/data/tetsu4/rgruhn/WSJ/nn` : experiments, readme-files, scripts and tools

Bibliography

[BGN02] Norbert Binder, Rainer Gruhn, and Satoshi Nakamura. Recognition of Non-Native Speech Using Dynamic Phoneme Lattice Processing. *Proc. Acoust. Soc. Jap.*, page 203f, 2002. spring meeting.

[CGN04] Tobias Cincarek, Rainer Gruhn, and Satoshi Nakamura. Cluster-based adaptation of acoustic models for non-native speech recognition. *Proc. Acoust. Soc. Jap.*, page 181f, 2004. spring meeting.

[GMN02] Rainer Gruhn, Konstantin Markov, and Satoshi Nakamura. Probability sustaining phoneme substitution for non-native speech recognition. In *Proc. Acoust. Soc. Jap.*, pages 195–196, Fall 2002.

[MTY$^+$04] Nobuaki Minematsu, Yoshihiro Tomiyama, Kei Yoshimoto, Katsumasa Shimizu, Seiichi Nakagawa, Masatake Dantsuji, and Shozo Makino. Development of English speech database read by Japanese to support CALL research. In *International Congress on Acoustics*, volume I, page 554, 2004.

[Pal90] K.K. Paliwal. Lexicon-building methods for an acoustic sub-word based speech recognizer. In *Proc. ICASSP*, pages 729–732, 1990.

[PJ92] D.B. Paul and J.M.Baker. The design for the wall street journal based CSR corpus. In *Proc. DARPA Workshop*, pages 357–362, Pacific Grove, CA, 1992.

[SC99] Helmer Strik and Catia Cucchiarini. Modeling pronunciation variation for ASR: A survey of the literature. *Speech Communication*, 29:225–246, 1999.

[Tom00] Laura Mayfield Tomokiyo. Lexical And Acoustic Modeling of Non Native Speech in LVCSR. *Proc. ICSLP*, pages IV:346–349, 2000.

[UB99] U. Uebler and M. Boros. Recognition of non-native german speech with multilingual recognizers. *Proc. EuroSpeech*, 1999.

[vC01] Dirk van Compernolle. Recognition of goats, wolves, sheep and ...non-natives. *Speech Communication*, 35:71–79, 2001.

[WY93] P. Woodland and S. Young. The HTK tied-state continuous speech recognizer. In *Proc. EuroSpeech*, pages 2207–2210, 1993.

[Y+99] S Young et al. *The HTK Book*. Entropic Ltd, 1999.

[YO99] Seong-Jin Yun and Yung-Hwan Oh. Stochastic lexicon modeling for speech recognition. *IEEE Signal Processing Letters*, 6:28–30, 1999.