PRONUNCIATION MODELLING AND LEXICAL ADAPTATION IN MID-SIZE VOCABULARY ASR

Louis ten Bosch, Nick Cremelie

Lernout & Hauspie Speech Products
Louis.tenBosch@lhs.be, Nick.Cremelie@lhs.be

ABSTRACT

A computational-phonological method is presented to automatically adapt the phone transcriptions in a lexicon to improve ASR performance in a number of mid-size recognition tasks. The lexical adaptation approach is based on supervised phoneme loops using cd-HMM segments to find alternatives for the transcriptions, and can be considered as a counterpart of the K-means algorithm but on symbolic level. The word error rate in a limited task (digit string recognition) with dialect speakers is shown to drop by 20-25 percent relative, starting from non-dialect digit transcriptions. Since the method is computationally involving, it is only feasible for relatively small tasks.

1. INTRODUCTION

One of the important difficulties for ASR is the variability of the pronunciation of words due to disfluencies, word contexts, speakers, speaking rates, speaking styles, and dialects/accents. In order to cover this variability, the recognition is often based on a pronunciation dictionary with more than one pronunciation variant per word (recognition token). Many studies have shown that a proper pronunciation modeling (PM) can improve the performance results, although the improvements are not always spectacular (see e.g. Strik & Cucchiarini, 1999). In a number of cases, however, the ASR performance may gain from a proper PM, for example spontaneous speech, and dialect and non-native utterances. For spontaneous speech, McAllaster et al. (1998) have shown (on simulated speech data) that word error rates on the Switchboard corpus decreased from 40 to 8 percent if the dictionary pronunciations matched the actual pronunciations. For a 50 k word dictation task, when sufficient data are available and a retraining of the acoustic models is possible, the gain is often limited. The reason is that, at least for languages such as English, French and Dutch, especially the (short) function words are prone to pronunciation variation, while PM usually generates more variants as the words become longer, which is from an ASR point of view less relevant.

In this paper, we show that, in the case of dialect speech, pronunciation modeling helps to substantially improve the recognition rate in small to mid-size tasks (the language model being based on a finite state grammar).

Pronunciation modeling uses techniques which can either be based on knowledge (knowledge about the language, dialects, phoneme rules etc.), be entirely data-driven, or a mix of both. The core issue in pronunciation modeling is the trade-off between the *generation* of promising pronunciation variants and the *selection* of appropriate variant candidates (c.f. Byrne et al., 1998; Cremelie & Martens, 1999, Riley et al.; 1999; Yang & Martens, 2000; Cremelie & ten Bosch, 2001). The generation of variants (knowledge-based or data-driven) usually improves the phonetic modeling of a word itself but it evidently might increase the inter-word confusability in the lexicon. The selection step aims at keeping this confusion within acceptable limits. Also the selection step can be knowledge-based or data-driven.

In the sequel, we will focus on the data-driven pronunciation modeling. We distinguish the following sub-steps:

1. a phone graph was created to allow a supervised phone loop in which deviations from the canonical transcriptions are penalized
2. variant generation: for each acoustic token in the acoustic database a 'surface' transcription was generated, by aligning with the phone graph
3. variant selection: selection of the set of candidate transcriptions for all words

In the next sections, these steps will be explained in more detail on the basis of a specific ASR experiment. Since the method in its current form is computationally expensive, it can be applied to mid-size vocabulary tasks with a finite state grammar.

2 LEXICAL ADAPTATION EXPERIMENT

The experiment is dealing with the recognition of digit strings that are all spoken in a dialect that is quite remote from the standard pronunciation on which the acoustic models in the recognizer are based. It aims at the automatic improvement of the recognition lexicon in order to improve the ASR performance.

Data

Loggings from a command and control task were used to select digit strings. All speakers spoke a local West-Flemish dialect. The recordings, done with a head set and a close-talk microphone, were selected so as to balance the variety of speakers and the number of digit tokens in the test material. In total 3200 utterances had been selected, each containing 2 or 3 digits. Data were recorded at 16 kHz and resampled to 11 kHz. The ASR system is based on a discrete acoustic model and was using triphone HMMs (no word models).

Using single canonical (standard Flemish) transcriptions for the ten digits 0 to 9, the overall performance for this task on a held-out test set is 86 percent. The main digit confusions are (in terms of P(recognized | uttered)): P(4|5) = 0.10, P(4|7) = 0.20, and P(1|9) = 0.31. The 1 is often inserted in the case a '9' is uttered. This overall poor recognition result is due to the very dialect pronunciations of the digits. The most marking dialect pronunciation was the use of high vowels in the utterance for '9' and the reduction of a syllable (an auditory check suggested the monosyllabic transcription /ni:n/, rather than standard disyllabic Flemish /nex$n/, where $ refers to the shwa). The ASR performance can easily be improved by constraining the digit string length in the input or by using word models rather than explicit phoneme-based models. However, the aim for using this data was to investigate how pronunciation modeling is capable to improve the phone transcriptions of all digits to reduce the error rate.

Variant generation, phone loop constraints

Since unsupervised phone loops usually overgenerate surface forms, a number of restrictions and pruning mechanisms have been suggested (e.g. association strength between segments: Amdal et al. 2000; reference-based constraints in the phone loop: Yang & Martens, 2000). In this research, we apply a phone loop which is biased towards the canonical solution, but allows deviations from it with a certain cost, by setting penalties (>0) on all arcs that are not included by the canonical (triphone) transcriptions. In contrast with Yang & Martens (2000) and Garcia et al. (1999), we use context-dependent HMM models in the phone loop. The graph contains about 4500 triphone models and about 65000 directed arcs. The resulting graph was containing all context-respecting directed arcs of triphone models, including a number of entry arcs and a number of exit arcs, in a similar way guided by the context of the triphones. The following table gives an impression of the number of deviations from the transcription of '9' /n'ex$n/ in terms of this arc penalty after forced alignment. A penalty value of 300 corresponds to the average score of a HMM-speech-state on a speech-frame. The total number of '9'-tokens used in this test is 80.

Arc penalty	Number of times /n'ex$n/ was found as 1-best	Number of different other transcr. found
1000	80 (100%)	0
600	70	7
500	57	14
400	35	28
300	17	43
200	6	54
100	0	70
0	0	77

For an arc penalty of 600, the following 7 different transcriptions were found: /n'e'I/, /n'em/, /n'en/, /n'enK/, /n'i$n/, /n'ii/, /n'in/. All these examples show the tendency to drastically reduce the second syllable in /n'ex$n/, thereby provoking the confusion with '1' /'en/. Because the transcription of '9' moves towards that of '1', the transcription of '1' has to shift a bit in order to attain sufficient discriminative power between '1' and '9'.

Since the phoneme loop syntax is rather CPU costly, we have been experimenting with a number of different, cheaper designs for the phoneme loop. One of the candidate designs that has been tested is a graph build with the preferred transitions as starting point. From the set of all transitions that respect the phoneme-context, (a) we selected the ones occurring in the preferred transcriptions (b) added all arcs that could context-wise possibly start from any arc selected in the previous step, and (c) iterated step (b) 4 times. Four times is enough to model substitutions, 1-phoneme insertions, and certain deletions. Unfortunately the resulting graph is not much smaller that the full original one used in the test described.

There is a parallel in the field of 'computational phonology'. In a paper on computational comparison and classification of dialects, Nerbonne & Heeringa (1999) used the Levenshtein distance – a measure of string distance – between narrow transcriptions, to define 'phonological distances' between regions (see ten Bosch, 2000). On one hand, they showed that the Levenshtein distance is a reasonable measure to quantify dialect distances. On the other hand, the way how deviations from the reference path are weighted by this Levenshtein distance is very similar to the way deviations are priced in the phone loop described above.

Variant selection

The selection of pronunciation variants can be based on e.g. rule firing probability (cf. Yang & Martens, 2000) or on the basis of log-likelihood of the variants (cf. Printz & Olsen, 2000), or on the basis of intra-lexicon confusion via word-word confusion (Riley & Roe, 1994; McAllaster et al., 1998; Printz & Olsen 2000; Garcia et al. 1999). In this study, we applied a strategy to define a set of winning transcription candidates by selecting the so-called Condorcet winners (see appendix). This selection strategy can be explained as follows. Assume that the N-best solutions for some tokens of X1 and X2 are distributed as in the following table:

1-best	Tr2-X1	Tr1-X1	Tr2-X2	Tr1-X3
2nd best	Tr1-X3	Tr5-X2	Tr4-X2	Tr4-X2
3rd best	Tr2-X2	Tr9-X2	Tr1-X1	Tr3-X2
...	Tr1-X1	Tr2-X1	Tr2-X1	...
...	...	...	...	...
reference	X1	X1	X1	X2

where Tri-Xj denotes transcription i associated with Xj. Two of the 3 tokens of X1 are correctly recognized, the token of X2 is recognized as X3. The problem of selecting the optimal set of transcription candidates for all X is now reduced to selecting the transcription candidates as if the table were a voting tableau such that all voters are maximally satisfied. The selection procedure is as follows:

1. Eliminate all transcriptions that never occur as first-best solution.
2. For all tokens of X1, select the Condorcet winner of all transcriptions of X1. The Condorcet winner of any two transcriptions is the transcription that is most frequent highest in rank. In the example above, this is Tr2-X1 for X1. Do this for all X.
3. Eliminate all tokens of all X that are correctly recognized after selecting transcriptions in step (2)
4. Repeat step (2) and (3) on the remaining tokens.

Error-Driven Transcription Demotion

The full lexicon adaptation method, Error-Driven Transcription Demotion, relies on the use of an N-best DP in which specific arcs can be allotted an additional penalty. It has some similarities with Hypothesis Driven Lexical Adaptation (HDLA, Waibel et al., 2000) and can also be compared with a k-means method on symbolic level. In the following scheme, 'X' denotes an arbitrary digit.

1. Free recognition: A recognition was done with a grammar allowing a loop over all digits, such that the hypothesis and the reference are forced to be equally long. The lexicon contains the transcriptions to be improved. Optional silences are allowed between the digits.
2. Transcription generation part A: For each token of X that is correctly recognized in (1), select the transcription corresponding to the detected 1-best hypothesis and store this in a 'list of candidate transcriptions' for X.
3. Transcription generation part B: For each token of X that is incorrectly recognized in (1), use a supervised phoneme loop (keeping the context the same), do a backtrace on phoneme level and keep the 1-best phoneme transcription. Store this in a 'list of candidate transcriptions' for X. As an example, suppose '1 2 3' has been uttered and '1 2 4' was recognized. Then the syntax graph during this backtrace step is **begin** – [sil] – 1 -- [sil] – 2 – [sil] -- phoneme_loop – [sil] – **end**. As explained before, the phoneme loop has been supervised ('tuned') not to generate transcriptions that are too deviant from the canonical transcription.
4. Transcription ranking: All transcriptions for all X occurring in the candidate lists are put in the lexicon as parallel words, and an N-best on word-level is done to obtain, for each token of X, a ranked listing of *all* candidate transcriptions. The grammar used (in the example) is **begin** – [sil] – 1 -- [sil] – 2 – [sil] – parallel_transcriptions – [sil] – **end**. The alignment scores themselves are not taken into account.
5. Repeat 2-4 for all X.
6. Selection of the candidates, according to the procedure described in the section 'variant selection' above
7. Repeat (1) to (6), until the transcriptions do not change.

Using this algorithm on a held-out development set, it was possible to improve the overall recognition rate on the held-out test set from 85 to 89 percent. Furthermore P(4|5) decreased from 0.1 to 0.07, P(4|7) from 0.2 to 0.16, and P(1|9) from 0.31 to 0.22, which amounts overall to about 25 percent error reduction compared to the test using the canonical transcriptions. This lexical adaptation has been performed for all speakers in the development set simultaneously. The pronunciation dictionary changed as indicated in the following table. The results are shown for the canonical transcription and after the second iteration of step 7. (The third iteration did not show any improvement).

digit	Canon.tr.	After it 2
0	n ^ l	n ^ l
1	'e n	e n I n
2	t w 'e	t w 'e
3	d r 'i	d r 'i
4	v 'I r	v I r f i r
5	v e&I f	v e&I f v 'I 'I f
6	z E s	Z E s
7	z 'e v $ n	z 'e v $ n z I I v $ n
8	A x t	A x t
9	n 'e x $ n	n 'e x $ n n 'e 'e n n 'i 'i n

3. FINAL REMARKS

The method proposed in this paper (error-driven transcription demotion) has proven its usefulness in a task with a small vocabulary using a finite state grammar. The word error rate could be reduced by about 25 percent, by modifying the phonetic transcription in the pronunciation dictionary. The

acoustic models have not been retrained. The improvement is a result of modifying the pronunciation transcriptions for all speakers at the same time. Not all speakers, however, uttered at the same dialect level. Some gain is expected by introducing pronunciation dictionaries that are dialect-level dependent.

The method to generate transcriptions is quite CPU expensive and relies heavily on the use of N-best dynamic programming. The search for alternative candidates is done via a supervised phone loop in which penalties are set on certain arcs. The graph is based on context dependent HMM segments. Currently we are looking at ways to increase the speed and extend the scope of the method.

The selection of pronunciation candidates is based on firing counts of transcriptions. Transcriptions are not ranked according to their first-best alignment scores, but rather to their ranking in an N-best list per acoustic token. This method works in the case of relatively small amount of tokens. The underlying assumption is that the ranking of N-best hypotheses per token does not change after removing a transcription candidate: for all transcriptions a, b, c: if $a < b < c$, then $a < c$ after removing b etc.

The method shows improvement on the number of substitutions, rather than on insertions and deletions. These errors can be dealt with in another way, e.g. by adjusting word transition penalties.

REFERENCES

1. Amdal, I., Korkmazskiy, F., Surendran, A. (2000). Data-driven pronunciation modelling for non-native speakers using association strength between phones. Proceedings ISCA workshop ASR-2000 , pp. 85-90.

2. Byrne, W., Finke, M., Khudanpur, S., McDonough J., Nock H., Riley M., Saraclar M., Wooters, C., Zavaliagkos, G. (1998) Pronunciation modeling using a hand-labeled corpus for conversational speech recognition. In: Proceedings ICASSP, pp. 313-316.

3. Cremelie, N., Martens, J.-P. (1999) In search of better pronunciation models for speech recognition, Speech Communication, vol. 29, p.115-136.

4. Cremelie, N., ten Bosch, L.F.M. (2001). Improving the recognition of foreign names and non-native speech by combining multiple grapheme-to-phoneme converters. Submitted to the ISCA-ITRW workshop on adaptation, Sophia Antipolis, France.

5. Garcia, P. Rubio, A., Diaz-Verdejo, J. Benitez, M., Lopez-Soler, J. (1999). A transcription-based approach to determine the difficulty of a speech recognition task. IEEE ASSP, vol. 7, 339-342.

6. Gehrlein, W.V. (1997). Condorcet's paradox and the Condorcet efficiency of voting rules. Mathematica Japonica. Vol. 45, pp. 173-199.

7. Greutner, P., Finke, M. and Waibel, A. (1998). Phonetic-distance-based hypothesis driven lexical adaptation for transcribing multilingual broadcast news. Proceedings ICSLP, Sydney, Australia, Nov. 1998. p. 2635-2638.

8. McAllaster, D., Gillick, L. Scattone, F., Newman, M. Fabricating conversational speech data with acoustic models: A program to examine model-data mismatch. ICSLP 1998, vol. 5, pp. 1847-1850.

9. Nerbonne, J., Heeringa, W., Hout, E. van den, Van der Kooi, P., Otten, S., Vis, W. van de (1999) Phonetic distances between Dutch dialects. (Manuscript in English).

10. Tajchman, G., Fosler, E., Jurafsky, D. (1995). Building multiple pronunciation models for novel words using exploratory computational phonology. Eurospeech 1995.

11. ten Bosch, L.F.M. (2000). ASR, dialects, and acoustic/phonological distances. ICSLP 2000, Beijing, China (cd).

12. Printz, H. and Olsen, P. (2000). Theory and practice of acoustic confusability. In Proceedings of the ISCA workshop ASR2000, pp 77-84.

13. Riley, M., Byrne, W., Finke, M., Khudanpur, S., Ljolje A., McDonough J., Nock H., Saraclar M., Wooters, C., Zavaliagkos, G. (1999) Stochastic pronunciation modeling from hand-labeled corpora. Speech Communication, vol. 29, pp. 209-244.

14. Strik H., Cucchiarini, C. (1999). Modeling pronunciation variation for ASR: A survey of the literature. Speech Communication, vol. 29, pp. 225-246.

15. Roe, D., Riley, M. (1994) Prediction of word confusabilities for speech recognition. ICSLP 1994, pp. 227-230.

16. Waibel, A., Geutner, P., Mayfileld Tomokiyo, L., Schultz, T., Woszczyna, M. (2000). Multilinguality in speech and spoken language systems. IEEE vol 88., no. 8, pp. 1297-1313.

17. Yang, Q., Martens, J.-P. (2000). On the importance of exception and cross-word rules for the data-driven creation of lexica for ASR, Proceedings ProRisk2000, pp. 589-593.

APPENDIX

The winner of a Condorcet election is the candidate who wins all pairwise match-ups. Condorcet's method is named after the 18th century election theorist (1743-94) who invented it. Condorcet's method and other pairwise methods let you rank the candidates in the order in which you would see them elected. The way the votes are tallied is by computing the results of separate pairwise elections between *all* of the candidates, and the winner is the one that wins a majority in *all* of the pairwise elections (Gehrlein, 1997).