TRANSITION PROBABILITIES ARE MORE IMPORTANT THAN WE ONCE THOUGHT

Guoli Ye, *Dongpeng Chen*, *Brian Mak*

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong

{yeguoli, dpchen, mak}@cse.ust.hk

ABSTRACT

It is generally believed that the transition probabilities in a hidden Markov model (HMM) have a limited role in the speech decoding process. In this paper, through a series of recognition experiments on Wall Street Journal (WSJ) read speech and SVitchboard (SVB) conversational telephone speech, we find that the HMM transition probabilities may be more important than we once thought. The experiments include: (1) setting or not setting all outgoing transition probabilities equal; (2) the introduction of word-final triphones and the re-estimation of their transition probabilities; (3) besides grammar factor and insertion penalty, the addition of a third decoding parameter called *transition factor* to scale the transition probability score during decoding. The results of the above three experiments enable us to improve the the word accuracy of the WSJ and SVB speech recognition task by 0.7% and 5.3% absolute respectively when compared to their baseline model in which all transition probabilities are simply set to 0.5.

Index Terms: transition probabilities, transition factor, phone deletion modeling, word-final triphones.

1. INTRODUCTION

It is a common belief that the transition probabilities in a hidden Markov model (HMM) have limited contribution during speech decoding. However, in our recent work on explicit modeling of phone deletions, when we construct what we call *context-dependent fragmented word models* (CD-FWM) in which skipping arcs are added to word models composed from well-trained triphones [1], we notice that the transition probabilities of the skipping arcs seem to matter. If we make all the outgoing transitions from a state of the CD-FWMs equal to each other and re-estimate the remaining model parameters, the recognition performance of the resulting model is significantly worse. The finding prompts us to re-visit the "common belief" again.

In this paper, we would present three experiments designed to study the contribution of the transition probabilities in a continuous-density hidden Markov model (CDHMM) for the recognition of read speech and spontaneous conversational speech. The experiments are conducted to answer the following three questions:

1. May we ignore the contribution of the transition probabilities and simply set them to 0.5 (since all CDHMMs in our experiments have the common strictly left-to-right topology with no skipping arcs)?
2. It is well-known that it is necessary to balance the dynamic ranges of acoustic score and language score by a grammar factor. Since the language score and the transition probability score are true probabilities (with a value between 0.0 and 1.0), if grammar factor helps, will the addition of a *transition factor* help too?
3. There are great variations in the realization of the coda of a syllable, especially in conversational speech [2]. [3] also shows that word boundary information may be utilized to improve Arabic speech recognition. Here we create additional word-final triphones (WFT) which differ from the their generic triphone counterparts mainly by the transition probabilities. Will WFTs improve speech recognition?

This paper is organized as follows. In Section 2, we first describe the two recognition tasks in this study, their corpus and experimental setup. The design of the experiment for each of the above three questions is then detailed in Section 3–5. Finally, Section 6 summarizes the findings and gives the concluding remarks.

2. RECOGNITION TASKS AND THEIR SETUP

Two speech recognition tasks are used for this investigation:

- WSJ: Wall Street Journal corpus [4] with a testing vocabulary of 5000 words. It consists of read speech.
- SVB: SVitchboard corpus [5] with a closed vocabulary of 500 words. It was extracted from Switchboard I [6]. It consists of spontaneous conversational speech.

In both tasks, acoustic vectors were extracted at every 10ms over a window of 25ms. Triphone models were then constructed using the HTK toolkit. All models are strictly left-to-right 3-state CDHMMs with a Gaussian mixture density at each state. In addition, there are a 1-state short pause model and a 3-state silence model. Finally recognition was performed again using the HTK toolkit with a beam width of 350.

All system parameters such as the decoding parameters and the state-tying tree were optimized using their development data set.

Table 1. Details of various Wall Street Journal data sets.

Data Set	#Speakers	#Utterances	Vocab Size	OOV
SI284	283	37,413	13,646	11.95%
si_dt_05.odd	10	248	1,260	0
Nov'93	10	215	1,004	0.29%

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

Table 2. Details of various 500-word Svitchboard data sets.

Set	#Speakers	#Utterances	#Words	Duration
train	324	13,597	51,324	3.69 hrs
dev	107	4,871	18,075	1.32 hrs
test	107	5,202	20,021	1.43 hrs

2.1. 5000-Word Wall Street Journal Task

2.1.1. *Speech Corpus*

The standard SI-284 Wall Street Journal (WSJ) training set was used for training the speaker-independent model. It consists of 7,138 WSJ0 utterances from 83 WSJ0 speakers and 30,275 WSJ1 utterances from 200 WSJ1 speakers. Thus, there is a total of about 70 hours of read speech in 37,413 training utterances from 283 speakers. All the training data are endpointed.

The standard Nov'93 5K test set with non-verbalized punctuations was used for evaluation using the standard 5K-vocabulary bigram that comes along with the WSJ corpus. The set si_dt_05.odd contains alternate sentences from the 1993 WSJ 5K Hub development test set after sentences with OOV words are removed. It was used to tune the system parameters. A summary of these data sets is shown in Table 1.

2.1.2. *Experimental Setup*

The traditional 39-dimensional MFCC vectors are used; they consist of 12 MFCCs and normalized log frame energy, and their first- and second-time derivatives. There are altogether 18,777 cross-word triphones based on 39 base phonemes. Each triphone state has a Gaussian mixture density of at most 16 components. There are totally 6,481 tied states which were derived from a phonetic decision tree.

2.2. 500-Word SVitchboard Task

2.2.1. *Speech Corpus*

SVitchboard (SVB) [5] is a conversational telephone speech corpus that is defined using subsets of the Switchboard-1 corpus [6]. It further defines several small vocabulary data sets ranging from 10 to 500 words, of which each task has a completely closed vocabulary. Each data set is further divided into five partitions, A – E, such that the speakers of one partition do not overlap with the speakers in any other partitions. In this paper, the Svitchboard 500-word task was used. Partitions A, B, and C were used for training; partition D was used for development, and partition E was used for testing. A summary of these data sets is shown in Table 2.

2.2.2. *Experimental Setup*

The number of base phonemes is originally 42 but it is reduced to 39 by converting [ax] to [ah], [el] to [ah l], and [en] to [ah n]. Thus, the baseline triphone system consists of 62,402 virtual triphones and 7,254 real triphones based on the final 39 base phonemes. There are totally 665 tied states, and each state has a Gaussian mixture density of at most 32 components.

The acoustic vector consists of 12 perceptual linear prediction (PLP) coefficients and the normalized log energy, as well as their first- and second-order derivatives. The lexicon provided by the Switchboard Transcription Project [7] was used. Finally, a bigram-backoff language model was constructed from the training data (partitions A–C) using the language modeling toolkit SRILM [8]. It has a perplexity of 36.4 on the test set.

Table 3. Results of Experiment 1. Recognition performance is measured in word recognition accuracy (%). (Figures with an $*$ are statistically and significantly better than other results in the same column.)

Model	WSJ	SVB
baseline1: set all $a_{ij} = 0.5$, train other parameters	91.35	41.36
baseline2: all HMM parameters are trained	91.40	44.26^*
model3: baseline2 pdf's, reset all a_{ij} to 0.5	91.31	41.65

Table 4. Comparison of triphone systems on the 500-word SVB test set. (WAC is word accuracy in %).

System	WAC
HTK; PLP [9]	38.80
GMTK [10]; PLP [9]	40.80
HTK; PLP (our baseline2)	**44.26**
GMTK; PLP + Fisher-trained, factored AF tandem [9]	46.20
GMTK; PLP + tuning on larger development set [11]	48.10

3. EXPERIMENT 1: EQUAL TRANSITION PROBABILITIES

Due to the common belief that the transition probabilities are not important in decoding, there are some practices of setting all of them to the same value, namely, 0.5 for a strictly left-to-right HMM that has no skipping arcs. Another reason that supports the practice is that the duration model deduced from the HMM transition probabilities is wrong.

For each of the two speech recognition tasks, we compare the following three models:

- baseline1: all transition probabilities a_{ij}'s are set to 0.5, and only the remaining HMM parameters are estimated.
- baseline2: all HMM parameters are estimated; this is always the base case in all experiments in this paper.
- model3: reset all transition probabilities a_{ij}'s in baseline2 models to 0.5 before using them for decoding. That is, model3 has the state pdf's of baseline2 model, but the transition probabilities of baseline1 model.

The performance of these three models is shown in Table 3. Before we discuss the results, we would like to remark that the SVB task is difficult if one is restricted to use only the resources available from the corpus, and thus the low recognition accuracies. Table 4 compares the results of various systems on the 500-word SVB task reported in the literature. One may see that our baseline2 result compares favorably among the top three systems that do not use data other than the SVB 500-word corpus. The 4th system made use of articulatory tandem features trained on the Fisher corpus [12], while the 5th system was tuned on a larger development set, of which the details were not described in [11].

The results show that the three models perform equally well in the read speech of WSJ, and this is no surprise to many of us as many of us have had the same observation in the past. However, we speculate that the phenomenon was mainly checked on read speech recognition and not on spontaneous conversational speech. From Table 3, the HMM transition probabilities actually matter for the recognition of conversational speech in SVB. There is a performance degradation between baseline2 model and baseline1 model when all transition probabilities are set to 0.5 in the latter. The worse performance of model3 further confirms that the loss is not due to the, perhaps, sub-optimal Gaussian pdf's in baseline1 model since model3 shares the same pdf's as the baseline2 model.

Table 5. WSJ Results of Experiment 2. (N = #tied states; WAC is word accuracy in %.)

Model	#HMMs	N	WAC
baseline2: trained a_{ij}	18,777	6,481	91.40
+ WFT	21,657	6,481	91.58
+ untying WFT states	21,657	7,562	91.71

Table 6. SVB Results of Experiment 2. (N = #tied states; WAC is word accuracy in %; the figure with an $*$ is statistically and significantly better than baseline2 result.)

Model	#HMMs	N	WAC
baseline2: trained a_{ij}	7,254	665	44.26
+ WFT	7,768	665	44.84
+ untying WFT states	7,768	767	45.12^*

4. EXPERIMENT 2: WORD-FINAL TRIPHONES

It is known that the actual realization of the same phoneme generally depends on the position of the phoneme in a word. Position-dependent monophones and triphones have been used for speech recognition in the past [13, 3]. Here we limited our investigation of word-final triphones (WFT) only for the following 4 phonemes: /t/, /d/, /s/, and /k/. One reason is that we postulate that the word-final plosives have more variations than other phones and in other parts of a word, and the effect is more pronounced in spontaneous conversational speech as they may not be well articulated before the next word starts. Another reason is that there are not sufficient WFT training samples for many other phonemes.

Below is the procedure for the construction of WFTs:

STEP 1: Modify the last phone in the pronunciation of each word in the dictionary using a new word-final phone label. For example, the pronunciation for the word "about" is modified from `/ah b aw t/` to `/ah b aw t:final/` where `/t:final/` is the word-final `/t/`.

STEP 2: Create new cross-word triphones that involve the new word-final phone. Note that if the same triphone context may appear at the end as well as other parts of a word, then the corresponding triphone model has to be cloned, and one of the duplicates is re-labeled with the new word-final phone and it now becomes a word-final triphone (WFT). Because all states are tied, the new WFTs will share the same state pdf's as the other triphones, but their transition probabilities will be updated separately.

STEP 3: Re-train all CDHMMs for another 8 EM iterations, both the state pdf's and transition probabilities.

STEP 4: For WFTs which have sufficient training data, untie their states and re-train all CDHMMs for another 8 EM iterations.

It turns out that for the WSJ task, there are enough training data for all observed WFTs of the four plosive phonemes, whereas for the SVB task, the database is so small that only the phoneme /t/ may have its own WFTs. We started with baseline2 model of Experiment 1, added WFTs to its triphone inventory, and checked the performance of the new models afterwards. The resulting model size, number of states, and recognition performance are summarized in Table 5 and Table 6 respectively for the two tasks.

It is observed that even with the simple addition of WFTs for four(one) phones improves the recognition performance of WSJ(SVB) by 0.31%(0.86%) absolute when compared with its baseline2 model. In the SVB case, the recognition gain provided by the WFTs of the single phone /t/ is statistically significant. More importantly, if we consider only the addition of WFTs without untying their states so that their state pdf's are the same as their non-word-final versions, and they only differ from their non-word-final versions in their transition probabilities, there are still 0.18% and 0.58% absolute gain in the recognition word accuracy of the WSJ and SVB task respectively.

5. EXPERIMENT 3: TRANSITION FACTOR

From the results of Experiments 1 and 2, it seems that the transition probabilities may play a small but significant role in speech decoding. If this is true, but since its dynamic range is much smaller than the acoustic likelihood, they should be properly weighted just like the language model score. Hence, in the third experiment, we investigate the addition of a third decoding parameter (after the grammar factor for language score, and word insertion penalty) that we call the *transition factor*.

We applied the large-margin iterative linear programming (LMILP) method in [14] for the discriminative training of the new transition factor. In fact, the grammar factor and word insertion penalty could also be estimated by LMILP. In [14], we showed that the grammar factor and word insertion penalty found by LMILP work at least as well as the ones found by an exhaustive grid search. Nevertheless, in this paper, only the transition factor was determined by LMILP, whereas the grammar factor and word insertion penalty were found by grid searches.

5.1. Review of LMILP

Details of the mathematical formulation for the LMILP training of the transition factor are similar to those for grammar factor and/or word insertion penalty, and the readers are referred to our paper [14]. In brief, LMILP is a discriminative training procedure for the unknown parameters in a linear function. In speech recognition, we would like to find the best word sequence such that

$$\hat{\mathbf{w}}_1^N = \underset{\mathbf{w}_1^N, N}{\operatorname{argmax}} \left\{ b(\mathbf{x}_1^T, \mathbf{w}_1^N) + K_{tf} a(\mathbf{x}_1^T, \mathbf{w}_1^N) + K_{gf} l(\mathbf{w}_1^N) + K_{wip} N \right\} ,$$

where $\mathbf{x}_1^T$ is the observation sequence; $\mathbf{w}_1^N$ is the decoded word sequence; $a(\cdot)$, $b(\cdot)$, and $l(\cdot)$ are the transition probability score, acoustic score, and language score respectively; K_{tf}, K_{gf} and K_{wip} are the decoding parameters and they are called the transition factor, grammar factor and word insertion penalty respectively. During LMILP, an N-best list is generated for each training utterance so that linear inequality constraints, which require the correct hypothesis to have greater recognition score than its competitors, may be derived. Obviously this cannot be guaranteed for all competitors and slack variables are introduced to the inequalities. Hence, the problem of finding the optimal decoding parameters is done through the minimization of the sum of these slack variables. For a more robust solution, a margin is added to the inequality constraints.

After a solution is found, a new N-best list may be generated using the new transition factor, and the procedure repeats until some convergence criterion is met.

5.2. Finding the Transition Factor by LMILP

Some details of the procedure are given below.

- The N-best hypotheses were generated by reserving 3 tokens at each state during Viterbi decoding.
- 20 best competing hypotheses were considered.
- All the utterances in the development set were used for LMILP training.
- The slack variables were tied for each utterance.
- The procedure started with $K_{tf} = 1.0$ and stopped when the change of its values between successive iterations was less than 0.01.
- The margin was set to a very large value, 10000. Basically, we want to set the margin to infinity so that each training data is effectively utilized in finding the optimal transition factor.
- The procedure was run on the models with WFTs in which the states were untied and re-trained.

Table 7. Results of Experiment 3. (WAC is word accuracy in %; figures with an $*$ are statistically and significantly better than baseline2 result.)

Model	WSJ	SVB
baseline1: all $a_{ij} = 0.5$	91.32	41.36
baseline2: trained a_{ij}	91.40	44.26
+ WFT	91.58	44.84
+ untying WFT states	91.71	45.12^*
+ transition factor	92.02^*	46.66^*

The optimal transition factors for the WSJ and SVB tasks are found to be 3.60 and 2.55 respectively. The ensuing recognition performance results are shown in Table 7 together with the results from the last two experiments.

The use of the transition factor to scale the transition probability score in the hypothesized word sequences gives an addition performance gain of 0.31% and 1.54% absolute in the WSJ and SVB task over their best model after the addition of WFTs and untying the WFT states. Moreover, the recognition improvement in the SVB task is more pronounced than that in the WSJ task.

6. SUMMARY AND CONCLUSIONS

This work presents three experiments to investigate the importance of the transition probabilities for speech recognition. The results show that the transition probabilities may give a small but significant contribution. From the experiments, it is advised to jointly estimate the transition probabilities together with other HMM parameters instead of simply setting them to 0.5. Then during decoding, a transition factor may be added to scale the transition probability score to further enhance the recognition performance. In our study, compared with the baseline1 model in which all transition probabilities are set to 0.5 before HMM training, the use of the re-estimated transition probabilities, word-final triphones, and transition factor together gives about 8% and 9% word error rate reduction on the recognition performance of the 5000-word WSJ task and 500-word SVB task respectively.

We caution that SVB is a small though difficult task, and advanced recognition techniques and external data are not employed in our studies. Further investigation is required to confirm the current findings when they are applied in larger speech corpora.

7. REFERENCES

[1] T. Ko and B. Mak, “Improving speech recognition by explicit modeling of phone deletions,” in *Proc. of ICASSP*, 2010.

[2] S. Greenberg, “Speaking in shorthand — a syllable-centric perspective for understanding pronunciation variation,” in *Proc. of the ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, 1998.

[3] F. Diehl, M. J. F. Gales, X. Liu, M. Tomalin, and P. C. Woodland, “Word boundary modelling and full covariance Gaussians for Arabic speech-to-text systems,” in *Proc. of Interspeech*, 2011, pp. 777–780.

[4] D. B. Paul and J. M. Baker, “The design of the Wall Street Journal-based CSR corpus,” in *Proc. of the DARPA Speech and Natural Language Workshop*, 1992.

[5] S. King, C. Bartels, and J. Bilmes, “SVitchboard 1: Small vocabulary tasks from Switchboard 1,” in *Proc. of Interspeech*, 2005.

[6] J. Godfrey, E. Holliman, and J. McDaniel, “Switchboard: Telephone speech corpus for research and development,” in *Proc. of ICASSP*, 1992, pp. 517–520.

[7] S. Greenberg, “The Switchboard transcription project,” in *Continuous Speech Recognition Summer Research Workshop Technical Report Series.* CLSP, JHU, USA, 1997.

[8] A. Stolcke, “SRILM an extensible language modeling toolkit,” in *Proc. of ICSLP*, 2002.

[9] K. Livescu *et al.*, “Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU summer workshop,” in *Proc. of ICASSP*, 2007.

[10] J. Bilmes and G. Zweig, “The graphical models toolkit: An open source software system for speech and time-series processing,” in *Proc. of ICASSP*, 2002.

[11] C. D. Bartels and J. A. Bilmes, “Graphical models for integrating syllabic information,” *Computer Speech and Language*, vol. 24, no. 4, pp. 685–697, Oct 2010.

[12] C. C. David, D. Miller, and K. Walker, “The Fisher corpus: a resource for the next generations of speech-to-text,” in *Proc. of LREC*, 2004.

[13] B. Mak and M. H. Siu *et al.*, “PLASER: Pronunciation learning via automatic speech recognition,” in *Proc. of HLT-NAACL*, 2003.

[14] B. Mak and T. Ko, “Automatic estimation of decoding parameters using large-margin iterative linear programming,” in *Proc. of Interspeech*, 2009, pp. 1219–1222.