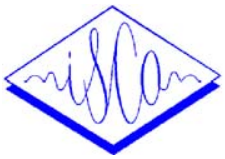
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

A Memory Efficient Grapheme-to-Phoneme Conversion System for Speech Processing

Jun Huang, Lex Olorenshaw, Gustavo Hernández-Ábrego, Lei Duan

Spoken Language Technology Group
Sony Electronics, USA
junh@slt.sel.sony.com, lex.olorenshaw@am.sony.com, gustavo@slt.sel.sony.com, lei@slt.sel.sony.com

10.21437/Interspeech.2004-455

Abstract

In this paper, a memory efficient, statistical data driven approach is proposed and successfully tested for grapheme-to-phoneme (G2P) conversion. In our system, a dynamic programming (DP) based fast algorithm is formulated to estimate the optimal joint segmentation between training sequences of graphemes and phonemes. A statistical language model is trained to model the contextual information between grapheme and phoneme segments. A two-stage fast decoding algorithm is also proposed to recognize the most-likely phoneme sequences given the input test word and the n-gram graphone models. Experimental results show that this system has similar recognition accuracy as a decision-tree based g2p system and requires much less memory and processing time.

1. Introduction

During the past decades, many different methods have been proposed for g2p conversion for the applications of dynamic speech recognition and text-to-speech (TTS) synthesis. These methods can be classified into two broad categories, namely the deterministic rule-based approach and the data-driven approach. In an entirely rule-based approach, the g2p rules are generated by phonetic/linguistic experts to find an exhaustive list of the mappings between graphemes and phonemes. This approach is very time consuming, language-dependent and not very accurate for some languages such as English. (One classic entirely rule-based g2p system carried out by researchers in the Naval Research Lab can be downloaded from [7].) The data-driven approaches can be further classified into three different sub-categories, namely inductive learning (IL), pronunciation by analogy (PbA), and graphone model.

In the IL approach, some statistical learning methods such as decision trees [5] were used to learn the g2p rules. The IL systems can generate reasonable phoneme accuracy (> 90%), but they usually require a lot of system memory (in the order of tens of megabytes in our experiments). The PbA systems adopt a more global consideration of the context information within grapheme sequences [1] in the process of g2p. However, this method also consumes a lot of system memory (again in the order of tens of mega bytes in our experiments) in order to build the pronunciation neighborhood for the entire training dictionary. Furthermore, the phoneme error rate is higher than that of the decision-tree based methods. A graphone is a pair composed by a letter sequence and a phoneme sequence of possibly different lengths [2,3]. In a graphone-based approach, some statistical modeling technique is used to learn the optimal alignment between the sequences of graphemes and phonemes, and the joint probabilities of each graphone.

Two examples of graphone-based g2p conversion systems were proposed in [2,6]. Our system differs from previous approaches in the following three major aspects: 1) our g2p system aims to provide more complete lexical information which includes not only the pronunciation, but also some prosodic information such as stress markers. 2) the major objective of our project is to design a multi-lingual g2p system with efficient memory consumption for potential application in electronic devices with limited system memory and processor speed. As a result, an efficient two-dimensional DP-based alignment algorithm is proposed to estimate the optimal joint-segmentation between the training word orthographies and their corresponding pronunciations, instead of using HMMs to obtain alignments as described in [6]. 3) in the decoding stage, a fast and memory-efficient, two-stage decoding algorithm is proposed in our system for g2p conversion, instead of directly using the standard A* decoding algorithm, which is widely used in large vocabulary continuous speech recognition (LVCSR) systems.

The rest of this paper is organized as follows. The descriptions of the statistical training of the unigram graphone model and the n-gram graphone model are given in section 2. A two-stage, fast decoding algorithm is presented in section 3. Some experimental results on our g2p system and the comparison with a baseline g2p system are presented in section 4. Finally, some discussion and conclusions are given in section 5.

2. Training of the graphone model

A graphone, or grapheme-phoneme joint multigram, is a pair comprised of a letter sequence and a phoneme sequence of possibly different lengths. For example, the word *rough* and its pronunciation / *r ah f* / can be represented by a set of three graphones, i.e., [*r, r*], [*ou, ah*], and [*gh, f*].

2.1 Training of the unigram graphone model

Let us define:

$$G \equiv \left\{ \vec{g}_i \right\}_{i=1}^{N} : \text{ set of training grapheme sequences}$$

$$\Phi \equiv \left\{ \vec{\phi}_i \right\}_{i=1}^{N} : \text{ set of training phoneme sequences}$$

where N denotes the size of the training pronunciation dictionary. A (m,n) graphone model is a model in which the longest size of sequences in G and Φ are m and n, respectively. For example, a *(4, 1)* graphone model means that one grapheme with up to 4 letters and one phoneme can be grouped together to form graphones. A joint segmentation (alignment) of $\vec{g}_i$ and $\vec{\varphi}_i$ is given by:

$$\vec{q}_t = (q_1, q_2, \cdots, q_L) = ([\tilde{g}_1, \tilde{\varphi}_1], [\tilde{g}_2, \tilde{\varphi}_2], \cdots, [\tilde{g}_L, \tilde{\varphi}_L]) \quad (1)$$

where:

$$\begin{cases} (\tilde{g}_1, \tilde{g}_2, \cdots, \tilde{g}_L) = \vec{g}_i \\ (\tilde{\varphi}_1, \tilde{\varphi}_2, \cdots, \tilde{\varphi}_L) = \vec{\varphi}_i \end{cases} \quad (2)$$

$q_j \equiv [\tilde{g}_j, \tilde{\varphi}_j]$, j = 1,2,⋯, L are the graphones.

The unigram (m,n) graphone model parameter set Λ^* is estimated using the maximum likelihood (ML) criterion:

$$\Lambda^* = \arg\max_{\vec{q}_t \in S(\vec{g}_i, \vec{\varphi}_i)} p(\vec{q}_t | \Lambda) \quad (3)$$

where $S(\vec{g}_i, \vec{\varphi}_i)$ is the set of all possible joint segmentations of $\vec{g}_i$ and $\vec{\phi}_i$. The parameter set Λ^* is trained using the expectation-maximization (EM) algorithm. The EM algorithm is implemented using forward-backward method to avoid the exhaustive search of all possible joint segmentation of graphone sequences. A marginal trimming technique is used to eliminate the unigram graphones whose likelihood are less than a certain threshold. During marginal trimming, the thresholds gradually increase from an initial small value to a larger value during each iteration of the training.

2.2 Joint segmentation of graphone sequences

After we obtain the unigram graphone model Λ^*, for each $(\vec{g}_i, \vec{\varphi}_i) \in (G, \Phi)$, $i = 1,2,\cdots,N$, we compute the optimal alignment using ML criterion

$$\vec{q}_i^* = \arg\max_{\vec{q}_t \in S(\vec{g}_i, \vec{\varphi}_i)} p(\vec{q}_t | \Lambda^*) \quad (4)$$

The optimal graphone sequence $\vec{q}_i^*$ actually denotes the optimal joint segmentation (alignment) between the grapheme sequence $\vec{g}_i$ and the phoneme sequence $\vec{\varphi}_i$, given the current trained unigram graphone model.

2.3 Training of n-gram graphone model

After we obtain the optimal joint-segmentation of grapheme and phoneme sequences, an n-gram language model of graphones is constructed to model the contextual information between grapheme-phoneme sequences. For example, the grapheme *ough* can be pronounced as */ah f/*, */uw/*, and */ow/*, as in words r*ough,* thr*ough*, and thor*ough,* respectively, depending on the context. In our system, we use Cambridge/CMU statistical language model (SLM) toolkit to train the n-gram graphone model. The priority of different backoff paths for tri-gram graphones is listed in Table 1.

Table 1. List of different backoff paths for the tri-gram graphone model

Priority	Approximation
5	$P(C \mid A, B)$
4	$P(C \mid A, B) * BO_2(A, B)$
3	$P(C) * BO_1(B) * BO_2(A, B)$
2	$P(C \mid B)$
1	$P(C) * BO_1(B)$

In table 1, $BO_2(A,B)$ and $BO_1(B)$ denote the backoff weights of trigram and bigram, respectively. In the decoding stage, the decoder always looks for the existing approximation of the n-grams with highest priority.

3. Decoding system

The basic idea behind the g2p decoder is to search the phoneme sequences that maximize the joint probability of graphone sequences given the input word orthography sequence $\vec{g}$:

$$\vec{\varphi}_{\vec{g}}^* = \arg\max_{\vec{q}_t \in S(\vec{g}, \varphi_t), \varphi_t \in S_p(\vec{g})} p(\vec{q}_t \mid \Lambda_{ng}) \quad (5)$$

where $S_p(\vec{g})$ denotes all possible phoneme sequences generated by $\vec{g}$, and Λ_{ng} denotes the n-gram graphone model. The joint probability of a graphone sequence given an n-gram graphone model can approximately be computed as follws:

$$p(\vec{q}_t) = p(q_1 \cdots q_L) = \prod_{i=1}^{L} p(q_i | q_1 \cdots q_{i-1}) \approx \prod_{i=1}^{L} p(q_i | q_{i-n+1} \cdots q_{i-1}) \quad (6)$$

In this paper, a fast, two-stage stack search algorithm is proposed to find the optimal pronunciation given the criterion described in Eq. (5). The decoding algorithm consists of two steps which can be described as follows.

Step-1: For an input word orthography sequence $\vec{g}$, search for the most likely grapheme segmentation of the input word. The basic idea is to find the segmentation with furthest depth, while complying with the backoff priority defined in Table 1.

Let us define depth i as the current number of grapheme segments, and $\vec{g}_{i+1,i+2,\cdots i+n}$ as the n-gram grapheme sub-sequences at current depth i; gs_i is the stack containing all possible grapheme segments at current depth i. The algorithm can be summarized as follows:

while (not_ end_of_word) do
construct all possible valid n-gram grapheme sequences $\vec{g}_{i+1,i+2,\cdots i+n}$ *based on the elements of previous stacks and n-gram graphone model*
if ($p(g_{i+n}|g_{i+n-1,\cdots,g_{i+1}})$ *exists) then*
push $\vec{g}_{i+1,i+2,\cdots i+n}$ *into* gs_i
else
search for backoff paths with the priorities described in table 1; construct the new valid backoff n-gram grapheme sequences, and push them into gs_i.
i++;

For example, consider the word *thoughtfulness*. The optimal segmentation after step 1 of the decoding algorithm, for a (4,1) graphone model with 3-gram SLM, is given by {*th, ough, t, f, u, l, n, e, ss*}.

Step-2: Given the optimal grapheme sequences resulting from step 1, search for the optimal phoneme sequences which will maximize the joint probability of graphone sequences defined in Eq. (6). Let us define n_{seg} as the number of grapheme segments resulting from step-1, n_g as the order of n-gram; $\vec{g}_i$ as the i-th n-gram grapheme in the grapheme stack, $\vec{\varphi}_{ij}$ as all possible n-gram phoneme sequences for grapheme $\vec{g}_i$; q_{ij} denotes the graphone constructed by grapheme $\vec{g}_i$ and phoneme sequence $\vec{\varphi}_{ij}$; ps_i denotes the stack of current phoneme candidates at depth *i*.

for $i \leftarrow 1$ *to* n_{seg} *do*
construct $\vec{g}_i = \{g_{i-n_g+1},\cdots,g_i\}$
find all possible $\vec{\varphi}_{ij}$ *from* Λ_{n_g}, *construct* q_{ij}
for $k \leftarrow 1$ *to* $|\vec{\varphi}_{ij}|$ *do*
for $l \leftarrow 1$ *to n do*
insert new phoneme token into ps_i
for each $q_{i+1,k}$ *allowed to follow* q_{ij} *do*
update the graphone stack and the likelihood of each graphone sequence in the stack
if ps_{i1} *is unique then*
pop out ps_{i1}
else
pop out the phoneme candidate with highest likelihood in the graphone stack; prune the stack

Let us assume the average length of the word orthography and the average number of phoneme mappings for each grapheme are *M* and *N*, respectively. For each input word, the number of possible grapheme segmentations is exponential to the word length. Furthermore, each grapheme can map to multiple phoneme entries in the pronunciation space, with different likelihoods. As a result, the computing and storage cost for direct solution of the search problem defined in Eq. (5) is at the order of $O(c_1^M)*O(c_2^N)$. On the other hand, step 1 of our decoding algorithm only requires *O(M)* number of operations. The step 2 of our decoding system requires $O(N^{n_g})$ operations, which is a nondeterministic polynomial (NP) problem. One feature of our decoder algorithm is that it reduces a two-dimensional exponential search problem into two one-dimensional NP search problems, while still keep the approximate optimization of Eq. (6).

4. Experimental results

4.1 Experiment setup

The experiments were conducted using SONY's internal English pronunciation lexicon, which contains 66,000 entries. Each entry contains one word orthography and its base-form pronunciation, including primary and secondary stress markers, and syllable boundaries. A random set of 56,000 entries was chosen for the training, and a disjoint set of 10,000 words was set aside for evaluation. We also compare the performance of our g2p system with a previous decision tree based g2p system described in [5]. The performance is measured in three categories, namely phoneme accuracy, memory consumption, and processing time.

4.2 Experimental results on the training of graphone model

In the training of the unigram graphone model, a marginal trimming technique was applied to eliminate outlier or rarely occurring graphones, and to significantly reduce the model size. Table 2 shows the size of unigram graphone models before and after the EM training, under different experimental conditions, where (m,n) denotes the order of unigram graphone model; $ev_{\min}$ and $ev_{\max}$ denotes the minimum and maximum threshold f evidence trimming, respectively. We can see from Table 2 that marginal trimming significantly reduces the number of unigram graphones, which can be viewed as a fundamental unit to represent the mapping between grapheme and phoneme sequences.

The trained unigram graphone model is used to generate alignments between words and their corresponding pronunciations in the training dictionary. This alignment data is used to generate n-gram graphone models in the second stage of training. Table 3 shows the experimental results on the size and perplexity of different n-gram graphone models. The first three rows in Table 2 denote the experimental results on trigram graphone model, and the last three rows denote the results on bigram graphone models. The cutoff threshold is set to 1 for all n-gram trainings

Table 2. Size of the unigram graphone model set before and after training using marginal trimming.

(m,n)	$ev_{\min}$	$ev_{\max}$	Size of graphone model set	
			Before	After
(2,2)	1e-6	1	106233	1641
(2,2)	1e-6	1e-1	106233	3326
(4,1)	1e-6	1e-2	78417	1931
(4,1)	1e-6	1e-3	78417	2028
(3,3)	1e-6	10	956513	908
(3,3)	1e-6	1	956513	5296

Table 3. Experimental results on n-gram graphone model.

(m,n)	ev_{min}	ev_{max}	Size of graphone model			Per -plexity
			1-gram	2-gram	3-gram	
(4,1)	e-6	e-2	1777	8342	28954	15.94
(2,2)	e-6	e-1	3189	21818	37702	42.64
(2,2)	e-6	e0	3097	22531	38434	43.11
(4,1)	e-6	e-2	1777	8342	N/A	27.34
(2,2)	e-6	e-1	3189	21818	N/A	79.51
(2,2)	e-6	e0	3097	22531	N/A	81.22

We can see from table 3 that we can use a rather small set of unigram ($O(10^3)$) and 2-gram or 3-gram ($O(10^4)$) graphones for English. Since the size of the n-gram models are rather small, this also enables a fast and efficient decoding algorithm for g2p conversion for different applications of speech processing. Another observation is that the perplexity of the graphone model almost doubled as we reduced from the tri-gram model to the bi-gram model.

4.3 Evaluation results on phoneme accuracy and memory consumption

In the evaluation stage, we measure the recognition accuracy, memory requirement, and the processing time of our new graphone based g2p system, and compare it with a decision tree based g2p system (baseline) described in [5]. Our g2p system is able to automatically generate a pronunciation dictionary with different options, such as both primary and secondary stress markers (for TTS) or without any stress marker (for dynamic speech recognition). Our system can also generate baseform pronunciations in SAMPA-IPA format or CMU phoneme format. In the training and decoding of the g2p system, vowels with different stress markers are treated as different phonemes, i.e., */ow/*, */ow1/*, and */ow2/*. This results in 71 different English phonemes with 26 consonants and 45 vowels with different stress markers. In our experiments, the unigram and n-gram graphone models are trained with labels of both primary and secondary stress markers. All of the evaluation experiments were conducted on the same PC with a Pentium 4, 2GHz CPU, and Linux Redhat 9.0 OS. The memory consumption and the processing time are measured for the entire 10k testing vocabulary.

Table 4 compares the performance of our graphone-based g2p system under different experiment conditions versus the baseline g2p system described in [5]. Here *#str* denotes the number of stress markers, *acc* means the phoneme accuracy, *corr* means the word correct, *mem* denotes the memory consumption for the decoding of entire 10k testing words, and *spd* means the processing time in seconds. We can see from Table 4 that our g2p system has similar recognition accuracy as the baseline. On the other hand, it only consumes 1/9 to 1/15 memory and is around 10 times faster when compared to the decision tree based system. We can also see from Table 4 that additional stress markers will increase the phone error rate by 3% - 4%, although further automatic labeling of secondary stress markers will only marginally increase the phoneme accuracy by less than 1%.

Table 4. Performance of graphone based g2p system.

(m,n)	ev_{min}	ev_{max}	*#str*	*acc (%)*	*corr (%)*	*mem (MB)*	*Spd (sec)*
(4,1)	e-6	e-2	0	91.3	53.4	3.56	12
			1	87.8	44.7		
			2	87.5	44.1		
(2,2)	e-6	e-1	0	90.5	49.8	6.34	9
			1	87.2	41.7		
			2	86.9	41.1		
baseline			1	88.4	50.1	55.6	134

5. Conclusion

In this paper, an efficient, data-driven approach was proposed for g2p conversion. Our g2p system is capable of obtaining similar phoneme recognition accuracy while requiring around 1/10 of system memory and 1/10 of processing time when compared against a decision tree based g2p system. There are several reasons why our g2p system is more efficient than the inductive learning based approach. First, instead of automatically learning many redundant rules, we obtained a compact representation of the internal mapping between the grapheme and phoneme sequences using graphone model. The size of the graphone model was significantly reduced through marginal trimming during the training stage. Second, instead of directly using the existing search algorithm in LVCSR, such as the A* algorithm, an efficient, two stage decoding algorithm was proposed in our work.

6. Acknowledgements

We want to thank Erika Kobayashi, Tomoaki Nitta, and Makoto Akabane from the speech lab of SONY NCE, Tokyo for their sponsorship, and assistance on this project.

7. References

[1] Bellegarda J.R., "Unsupervised, Language-independent Grapheme-to-Phoneme Conversion by Latent Analogy," in *Proc. ICASSP*, pp. 244 – 247, Hong Kong, April, 2003.

[2] Biasani M. and Ney H., "Investigation on Joint-Multigram Model for Grapheme-to-Phoneme Conversion," in *Proc. ICSLP*, pp. 105 – 108, Denver Colorado, September, 2002.

[3] Deligne S., Yvon F., and Bimbot F., "Variable-Length Sequence Matching for Phonetic Transcription using Joint Multigrams," in *Proc. Eurospeech*, pp. 2243 – 2246, Madrid, Spain, September 1995.

[4] Fisher W., "A Statistical Text-to-Phone Function Using N-Grams and Rules," in *Proc. ICASSP*, Phoenix Arizona, March 1999.

[5] Schweitzer A., Rapp S., Olorenshaw L., and Schimanowski J., "Automatic Pronunciation Generation for Speech Recognition," in *Proc. SRF*, 2001.

[6] Vozila P. *et al.,* "Grapheme to Phoneme Conversion and Dictionary Verification using Graphonemes," in *Proc. Eurospeech*, pp. 2469 – 2472, Geneva, Switzerland, September 2003.

[7] URL of Naval Research Lab text to phoneme rules, file://svr.ftp.eng.cam.ac.uk/pub/comp.speech/syntheses/english2phoneme.tar.gz