A Comparable Phone Set for the TIMIT Dataset Discovered in Clustering of Listen, Attend and Spell

Jialu Li
Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, US
jialuli3@illinois.edu

Mark Hasegawa-Johnson
Beckman Institute, University of Illinois Urbana-Champaign, US
jhasegaw@illinois.edu

Abstract

Listen, Attend and Spell (LAS) [4] maps a sequence of acoustic spectra directly to a sequence of graphemes, with no explicit internal representation of phones. This paper asks whether LAS can be used as a scientific tool, to discover the phone set of a language whose phone set may be controversial or unknown. Phonemes have a precise linguistic definition, but phones may be defined in any manner that is convenient for speech technology: we propose that a practical phone set is one that can be inferred from speech following certain procedures, but that is also highly predictive of the word sequence. We demonstrate that such a phone set can be inferred by clustering the hidden nodes activation vectors of an LAS model during training, thus encouraging the model to learn a hidden representation characterized by acoustically compact clusters that are nevertheless predictive of the word sequence. We further define a metric for the quality of a phone set (the sum of conditional entropy of the graphemes given the phone set and the phones given the acoustics), and demonstrate that according to this metric, the clustered-LAS phone set is comparable to the original TIMIT [5] phone set. Specifically, the clustered-LAS phone set is closer to the acoustics; the original TIMIT phone set is closer to the text.

1 Introduction

Traditional automatic speech recognition (ASR) usually is composed of multiple components including an acoustic model, a language model, and a pronunciation dictionary. Recently, modern ASR models implemented based on neural networks, such as connectionist temporal classification (CTC) [6] and LAS, have demonstrated the ability to learn a complete end-to-end optimal transformation from speech to text. Since such models generally are not dependent on utilizing specific language models or pronunciation dictionaries, their architectures are popular with new researchers trying to enter speech recognition community. Typical neural-network based models rely on the Recurrent Neural Networks (RNNs), and the key to success of utilizing such deep learning mechanisms is their ability to learn a hidden representation of the training data.

In this work, we take a step further to explore the possibility of defining a new phone set for the TIMIT dataset using the LAS model by incorporating a clustering method to soft align acoustics and graphemes. In the LAS model, the Listener takes the input acoustic signals and encodes the signals to a hidden nodes vector, the hidden nodes vector is accumulated with weights computed by an Attender, and then the accumulated hidden nodes vector feeds into the Speller to generate transcripts. Since the hidden nodes vector represents the relationship between the generated graphemes and the input

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

acoustic signals, we cluster the hidden nodes that are maximally attended by one or more output trigraphs. In this way, we train the model to learn the underlying relationship between the graphemes and acoustics.
The clustered pairs of hidden nodes and corresponding graphemes are the new defined phone set. We evaluate the new phone set by using an entropy utility function, the sum of the conditional entropy of the graphemes given the phones, and of the phones given the acoustics. The experiment reveals that the new phone set is comparable with the reference phone set. The new phone set has lower conditional entropy of phones given the acoustics than the reference phone set, but the new phone set has higher conditional entropy of graphemes given phones. Overall, the reference phone set is better.

2 Related Work

2.1 End-to-end learning in ASR

In order to learn the active speech-to-text transformation in an end-to-end fashion, it is necessary to represent several discrete optimization problems as differentiable continuous optimizations. Two approaches using RNNs have recently been successful: connectionist temporal classification (CTC) [6] and LAS. CTC generates the labels of a sequence of data with RNNs by estimating the probability distribution given the input sequence once per input time step, whereas LAS uses an Attender to accumulate state vectors across input time steps, and generates output characters at each output time step using a sequence-to-sequence mechanism given the transformed input nodes vector. Hybrids of CTC and LAS also exist [8]. Sequence-to-sequence neural network models with attention were first proposed for machine translation, and therefore some machine translation toolkits are able to implement LAS [12].

2.2 Representations between acoustics and text in speech recognition models

Deep learning works if and only if it's able to find an accurate hidden representation of training data, thereby enabling the system to learn the relationship between the input signal and output words. For conventional ASR, the phone is the smallest temporal unit in speech and serves as an intermediate representation connecting speech and text. A Hidden Markov Model (HMM) observes acoustic signal features and computes the likelihoods of triphone states, from which the likelihoods of words may be computed [9]. Hybrid HMM-(Neural Network)NN systems use the NN to compute phone likelihoods, and the HMM to compute phone alignment [11]. The transformation from speech to text always consists of a series of smaller transformations, from one hidden representation to the next, with each representation slightly more text-like and less speech-like than the one before it. In traditional ASR, these representations are explicit: triphone, phone, and word. In end-to-end models such as Deep Speech [1], the same sequence of transformations is learned from data: Belinkov and Glass [3] investigate the hidden representations of Deep Speech 2 [1], and the study shows that the phonetic information loss gradually increases from the bottom layer to the top layer.

3 Model

3.1 Brief descriptions of LAS model

LAS is an end-to-end speech recognition model that generates the transcripts directly from input acoustic signals without the implementations of multiple submodules of traditional ASR. The basic LAS model includes two modules: a Listener and a Speller. The Listener comprises three layer of pyramidal Bidirectional Long Short Term Memory (pBLSTM) [7], which encode the input acoustic signals and reduce the input time length to one-eighth of the original. The output of the Listener is represented by hidden nodes vectors $\mathbf{h}$, then the vectors are fed into the Speller as the input. The Speller is a sequence-to-sequence attention-based LSTM transducer. The attention mechanism of the transducer takes the hidden nodes vectors from the Listener and character distribution from previous step as input and generates a context vector as a weighted sum of all input hidden nodes vectors. The context vector is then used to generate the output character at the current step.

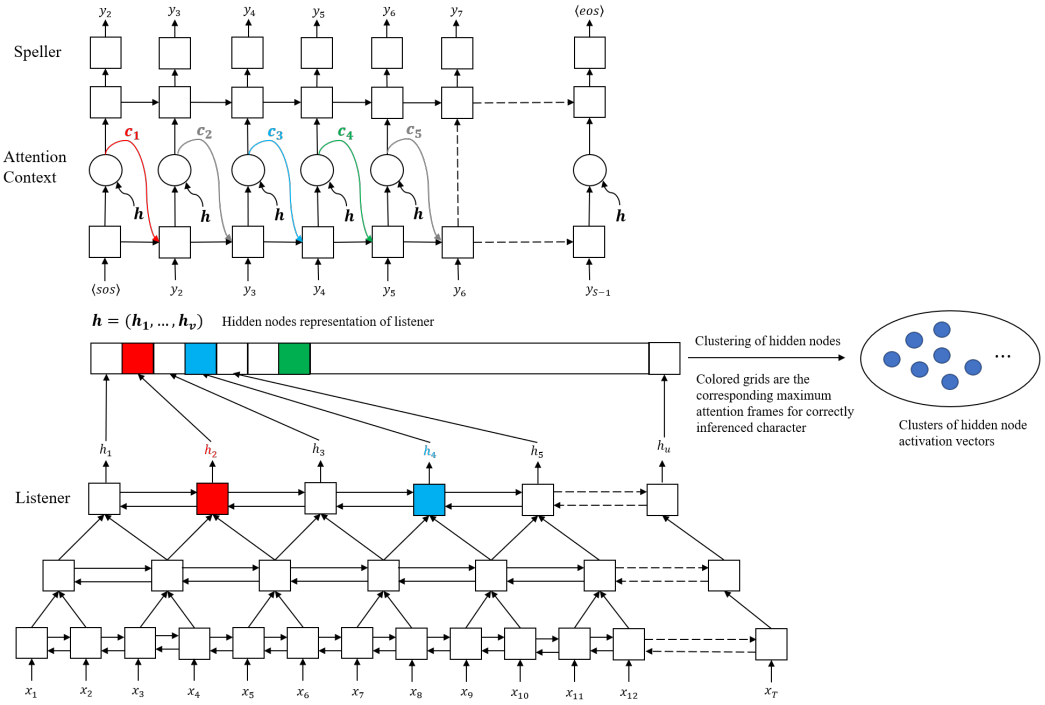

Figure 1: Modified architecture of Listen, attend and spell

The input of the clustering algorithm are the corresponding hidden nodes of the maximum attention frames generated by the AttentionContext vector for correctly inferenced character from the Speller. In this figure, y_2, y_4, and y_5 are correctly inferenced characters, and their corresponding context vectors of $\mathbf{c_1}$, $\mathbf{c_3}$, and $\mathbf{c_4}$ generate the attention vectors whose maximally attended input frames are h_2, h_4, and h_7 respectively. Thus, h_2, h_4, and h_7 are the input of the cluster.

3.2 Experiment LAS model

Figure 1 shows the overall modified LAS model of the experiment. We are aiming at finding the hidden relationship between the input acoustic signals and output transcripts, so we introduce a clustering component in the original LAS model to encourage the Listener to learn a hidden representation in which frames are grouped into compact clusters. Specifically, for each character correctly inferenced by the LSTM transducer from the Speller, we cluster the corresponding maximally attended hidden nodes. The hidden nodes vectors are a cumulative nonlinear transformation of the Mel-Frequency Cepstral Coefficients (MFCCs), and are trained to optimally summarize whatever information about the MFCC is necessary for the Speller to correctly generate output characters. By clustering maximum attended hidden nodes, we force the system to learn groupings of speech frames that have similar hidden nodes vectors and are also connected to similar output character sequences.

3.3 Learning

The modified LAS system can be trained jointly for accurate character output, but also for optimally clustered internal hidden nodes vectors. The training criterion of the modified LAS model contains two parts: word loss and clustering loss. The loss function can be described as the following,

$$\varepsilon = -\frac{1}{T_y}\sum_{i=1}^{T_y}\log P(y_i|\mathbf{x}, y_{1,(i-1)}) + \frac{\sum_{i=1}^{T_y}[y_i = \hat{y}_i]\|\mathbf{h_t}(i) - \mu_{k(t(i))}\|^2}{\sum_{i=1}^{T_y}[y_i = \hat{y}_i]}$$

$$t(i) = \underset{t}{\operatorname{argmax}}\,\alpha_{it} \quad k(t) = \underset{k}{\operatorname{argmin}}\,\|\mathbf{h_t} - \mu_k\|^2$$

where T_y is the length of generated output characters. α_{it} is the attention computed between output frame i and input frame t, and $[.]$ is unit indicator function. The first part of the training criterion is the

cross entropy between the reference transcripts and the generated transcripts, and is the error measure used in the standard LAS algorithm. $\mathbf{x}$ is the original input acoustic signal; y_i and $y_{1:(i-1)}$ refer to the reference character at output time i, and the sequence from times 1 through $(i-1)$. The second part is the average squared distance between hidden nodes attended by correctly inference characters and their closest cluster centroids. $t(i)$ is the index of the input frame that is maximally attended by output frame i. $\mathbf{h_t}$ is the hidden nodes vector at input frame t. The inferenced character at each output step is y_i, and corresponding reference character is $\hat{y}_i$. When the inferenced character is the same as the reference character, the corresponding maximally attended hidden node $\mathbf{h_t}(i)$ is selected to be the input of a clustering algorithm. $\mu_{k(t(i))}$ is the corresponding cluster centroid. $k(t)$ is the index of the closest cluster of hidden nodes vector $\mathbf{h_t}$. By minimizing this error function, we encourage the Listener to learn a hidden embedding, $\mathbf{h_t}$, that is useful in predicting the output character y_i, but that can also be clustered into compact phone-like clusters with centroids μ_k.

3.4 Clustering method

The clustering method in the modified LAS model is a standard k-means clustering algorithm except that the input varies, since the hidden nodes change for every batch during training. The objective of the clustering method is to minimize the clustering loss from the loss function.
The centroids of the clusters are randomly initialized with a normal distribution. The hidden nodes are clustered and labeled for a certain number of iterations for every epoch. Then the centroids are updated and kept for the next epoch. After every epoch, the empty clusters not used during the completed epoch will be deleted, and replaced by splitting the largest labeled clusters by scaling the original centroids by factors of 1.00 and 0.99.

4 Experiment

4.1 Dataset descriptions

Two datasets are used to perform the experiment. The English speech recognition training corpus of TED-LIUMv2(TEDLIUM) [15] is used to pre-train the LAS model. The TEDLIUM dataset was made from audio talks and transcripts from the TED website. There are 1495 audio talks with aligned transcripts in the dataset.
The TIMIT dataset is used to train the experimental LAS model. The TIMIT dataset comes with its self-defined dictionary and phoneme alignment transcripts for the audio talks. During training, audio and transcripts of one female and one male are randomly selected from each dialect region of the original TIMIT test dataset as development set, and the rest of the original TIMIT test dataset remains as evaluation test set. A new phone set is discovered for the TIMIT dataset and compared with the reference phone set.

4.2 Preprocessing of dataset

Both datasets are converted to MFCCs [10]. The raw acoustic signals in the dataset are multiplied by 25ms Hamming windows once every 10ms, and the sampling rate of the input signals is 16000 Hz. The power spectrum is calculated for each frame as the squared magnitude of Discrete Fourier Transform (DFT) of the original acoustic signals. Forty filters in Mel-spaced filterbank are applied to the power spectrum, and log filterbank energies are computed by taking the log of the power spectrum. Then the Discrete Cosine Transform (DCT) of these forty log filterbank energies gives the cepstrum coefficients.

4.3 Experimental settings

The implementation of the basic LAS model is based on the toolkit eXtensible Neural Machine Translation (XNMT) [13] using the Dynet framework [12]. The learning rate of the Adam optimizer is initialized to 0.01 and reduced to half of the original learning rate if WER of the development set isn't improved after 3 epochs. Other parameters are set to the values as indicated by the original paper [4]. The hidden dimension of pLSTM is 512, which is the dimension of the hidden nodes vector. The Attender has hidden dimension 128. The dropout rate of the entire neural network is 0.3. The Speller uses a beam search with size 20 to infer test transcripts.

The experimental model is modified by introducing a new clustering module. The LAS model has been pre-trained for about 300 epochs. Starting with the pre-trained model, the experimental model is then trained for 100 epochs; the learning rate of the Adam optimizer remains at 0.03. The number of iterations for each clustering step is 20, and the dimension of each cluster centroid is the same as the dimension of the hidden nodes vector, which is 512 in this case. The number of clusters is 100, which is roughly twice the number of unique English phonemes [2].

5 Results and discussion

5.1 Error measurements of experimental LAS model

Upon convergence, the pre-trained model of LAS has word error rate (WER) 16.72% and character error rate (CER) 8.46% on the test dataset. With the pre-trained LAS model, the experimental model has the final WER 26.99% and CER 10.67%. By the training criterion, clustering loss is 16.935 and maximum likelihood estimation loss is 0.948 per character.

5.2 Comparisons of new discovered phone set and reference phone set

For the experimental LAS model, 100 clusters are used to discover a new phone set for TIMIT. For all generated transcripts in the test set, every character y_i is assigned to the input state vector $\mathbf{h_t}$ according to $\mathbf{h_t}(i) = \operatorname{argmax} \alpha_{it}$, where α_{it} is the attention, and the closest μ_k of the corresponding hidden node $\mathbf{h_t}$ is credited with one occurrence of the label y_i. Conversely, the trigraph sequence $y_{1:T_y}$ is mapped to a phone sequence by looking up each word in the dictionary, then aligning the dictionary phones with trigraphs using Phonetisaurus. The top five most frequently assigned trigraphs for each cluster vote to determine the phone label of the cluster. Clusters with no clear majority phone label are reviewed, and the label is edited by hand. For example, the top five most frequently labeled trigraphs for one cluster are " wh", "ere", "whe", "wer", and " we". The cluster certainly captures the similar pronunciations of the center letters of the words "where" and "were", since lip rounding dominates both words, for this cluster, we assign the phone label as "w". The final phone set discovered by the clusters of the experimental system contains 40 unique phones.
We used both phonetic reference transcripts and the TIMIT dictionary for constructing two different phone sets. The reference phone set constructed from the phonetic transcripts contain all possible phonemic and phonetic symbols used in the TIMIT lexicon, and the total number of unique phones is 61 in the phonetic transcripts. Since the reference transcripts of the TIMIT dataset contain the actual pronunciations of the words, the phones of the transcripts are very different from the ones used in the TIMIT dictionary. We also try to find a phone set that represents the reference transcripts using the dictionary. We utilize the function phonetisaurus-align in toolkit Phonetisaurus G2P [14] to generate the alignment between each character and the corresponding phone of the reference transcripts. The stress markers of the TIMIT dictionary are eliminated. The reference phone set constructed from TIMIT dictionary contains 47 unique phones.
The experimental and both reference phone sets are as shown in the Table 1.

5.3 Entropy measurement

Entropy is commonly used to measure the randomness or disorder of a system. The output of the experimental model is evaluated by calculating the conditional entropy given different contexts for both the experimental and reference phone sets. The contexts include both phones and acoustics, and the dependent variables include phones and graphemes. The graphemes consist of all lowercase English letters and special tokens, including apostrophe, dash and space that appeared in both generated and reference transcripts. The acoustics include all individual frames in the test dataset from the TIMIT corpus. The conditional entropy is calculated as follows,

$$H(Y|X) = -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log p(y|x)$$

Table 1: List of phone sets discovered by experimental model and constructed from both reference dictionary and phonetic transcripts. The bolded phones in both second and the third column highlight the differences between the experimental and reference phone set.

phone categories	experimental model	reference (dict)	reference (phn)
stops	b, d, g, k, p, t	b, d, g, k, p, t	b, **bcl**, d, **dcl**, g, **gcl**, k, **kcl**, p, **pcl**, t, **tcl**, **dx**, **q**
affricates	ch, jh	ch, jh	ch, jh
fricatives	f, s, sh, th, v, z	**dh**, f, s, sh, th, v, z, **zh**	**dh**, f, s, sh, th, v, z, **zh**
nasals	en, m, n, ng	**em**, en, **eng**, m, n, ng	**em**, en, **eng**, m, n, ng, **nx**
semivowels and glides	el, hh, hv, l, r, w	el, hh, l, r, w, **y**	el, hh, hv, l, r, w, **y**
vowels	ae, ao, aw, ax, axr ay, eh, er, ey, ih, ix, iy, ow, oy, uw	**aa**, ae, **ah**, ao, aw, ax axr, ay, eh, er, ey, ih ix, iy, ow, oy, **uh**, uw	**aa**, ae, **ah**, ao, aw, ax, axr, **ax-h**, ay, eh, er, ey, ih, ix, iy, ow, oy, **uh**, uw, **ux**
non-speech event	h#	h#	h#, **pau**, **epi**

Table 2: Entropy of the distribution $P(\text{graphemes}|\text{phones})$, $P(\text{phones}|\text{acoustics})$, and $P(\text{graphemes}|\text{acoustics})$ for both experimental and reference phone sets. kNN and actual conditional distribution are used to estimate experimental conditional entropy.

System	$H(\text{gra}\|\text{pho})$	$H(\text{pho}\|\text{aco})$	$H(\text{gra}\|\text{aco})$
kNN-experimental model	1.959	0.424	1.198
kNN-reference (dict)	1.871	1.081	1.351
kNN-reference (phn)	1.757	1.131	1.348
experimental model	2.563	3.681	3.353
reference (dict)	0.776	3.842	3.355
reference (phn)	1.376	4.105	3.355

where X is context variable, and Y is the dependent variable.
For calculating all conditional entropy, we use both actual conditional distributions and estimated conditional distributions utilizing K-nearest neighbors (kNN) algorithm with K=10 for all contexts and dependent variables.
Specifically, when we use the actual distribution for calculating $H(\text{graphemes}|\text{phones})$,

$$p(x) = \frac{\text{number of occurrences of phone } x \text{ in transcripts}}{\text{total number of phones in transcripts}}$$

$$p(y|x) = \frac{\text{number of occurrences of phone } x \text{ aligned with character } y + \lambda}{\text{number of occurrence phone } x \text{ in transcripts} + \lambda \times \text{number of distinct characters}}$$

Laplace smoothing is applied for all conditional entropy calculations with the smoothing factor $\lambda = 1$.
Similarly, for calculating $H(\text{phones}|\text{acoustics})$, we also calculated the prior and conditional distributions. We approximate each individual frame as a unique frame. Thus, the prior acoustic distribution can be approximated by

$$p(x) \approx \frac{1}{\text{total number of acoustic frames in test set}}$$

$$p(y|x) = \frac{\text{number of occurrences of phone } y \text{ given acoustic frame } x + \lambda}{\text{number of output phones that attend acoustic frame } x + \lambda \times \text{number of phones in defined phone set}}$$

$H(\text{graphemes}|\text{acoustics})$ can be computed by following the same procedures of calculating $H(\text{phones}|\text{acoustics})$. $H(\text{graphemes}|\text{acoustics})$ differs slightly between the experimental and reference models, because each acoustic frame may be attended by more than one output character.

Phonetisarus G2P toolkit is also used to calculate the maximum likelihood alignments between the graphemes and acoustics in phonetic transcripts. Each sentence is treated as one word, and its corresponding phonetic transcripts are used train the finite-state-transducer model for the alignment calculations.

When we use the kNN algorithm, the prior is the same with the ones used in the actual distribution to calculate $H(\text{graphemes}|\text{phones})$. For both experimental and reference phone sets, we find all the phone centroids. For experimental phone set, for each phone centroid x with value of μ_k, we find the closest K vectors $h_t(i)$ by measuring the distance $\|h_t(i) - \mu_k\|^2$, that is, these $h_t(i)$ are from the set of hidden nodes vectors that are maximally attended by some grapheme. For reference phone set, we first calculate the phone centroid x as the average of all acoustic frames aligned with each phone. Then we find K closest frames of each phone centroid x aligned with grapheme y, so

$$p(y|x) = \frac{\text{number of occurrences of grapheme } y \text{ for phone centroid } x}{K}$$

Similarly, for calculating $H(\text{phones}|\text{acoustics})$, we find the corresponding hidden nodes vector h_τ for each acoustic frame x, then we find the closest K vectors $h_t(i)$ by $\|h_t(i) - h_\tau\|^2$, and these $h_t(i)$ are from all hidden nodes vectors that are used in the clustering algorithm for test dataset. The phone labels of each h_τ are used to estimate the conditional distribution,

$$p(y|x) = \frac{\text{number of occurrences of phone } y \text{ for acoustic frame } x}{K}$$

We follow similar procedures of calculating $H(\text{phones}|\text{acoustics})$ to calculate $H(\text{graphemes}|\text{acoustics})$. For each acoustic frame x, we find the closest K vectors $h_t(i)$ for h_τ from the set of hidden nodes vectors that are maximally attended by certain grapheme,

$$p(y|x) = \frac{\text{number of occurrences of character } y \text{ for acoustic frame } x}{K}$$

In order to measure the quality of the experimental phone set, we exploit Markov Chain property and have the inequality $H(\text{graphemes}|\text{phones}) + H(\text{phones}|\text{acoustics}) >= H(\text{graphemes}|\text{acoustics})$. From Table 2, when we use the actual distribution, the conditional entropy $H(\text{graphemes}|\text{acoustics}) \approx 3.355$ is similar for both experimental and reference phone sets. For the experimental system, $H(\text{graphemes}|\text{phones}) + H(\text{phones}|\text{acoustics}) - H(\text{graphemes}|\text{acoustics}) = 2.563 + 3.681 - 3.353 = 2.891$, meaning that quantizing to the experimental phone set would introduce 2.891 nats of entropy to the LAS speech recognizer. Similarly, the comparable number for dictionary reference phone set is 0.776+3.842-3.354= 1.264 and for phonetic reference phone set is 1.376+4.105-3.355=2.126. Compared with reference phone set, the experimental phone set is closer to acoustics but farther from the text. Overall, the reference phone set is slightly better. In particular, the reference (dict) phone set is very close to the text, because it is computed by looking up each word in the dictionary. The experimental phone set, conversely, is close to the acoustics, because it is created by clustering the hidden nodes vectors of the Listener in an LAS speech recognizer. Even though the experimental phone set is created from the Listener of the LAS model, however, it is not much farther from the text than the reference (phn) phone set, which is based on manual transcription of the pronunciations actually used by the speakers in the TIMIT dataset. This result suggests that the experimental phone set is capturing pronunciation variation comparable to the pronunciation variation represented by the reference (phn) transcriptions.

6 Conclusions

We defined a new phone set for the TIMIT dataset based on incorporating a clustering mechanism into the original LAS model. The learning criterion for the experiment model is composed of two parts: negative log probability of the reference transcripts, and squared distance between clustered hidden nodes and corresponding centroids of clusters. The learning criterion balances the learning objectives of the system – reducing the WER of generated transcripts meanwhile grouping the hidden vectors into compact clusters. The model is pre-trained with a larger dataset, TEDLIUM, and then trained on the TIMIT dataset for the experiment. The experiment result is evaluated by defining a utility function, the sum of the conditional entropy of graphemes given phones and the conditional entropy of phones given acoustics. By quantizing both experimental and reference phone sets, we show that the experimental phone set has higher "grapheme entropy" but lower "acoustic entropy". In conclusion, the experimental phone set is comparable with the TIMIT reference phone set, but TIMIT reference phone set is superior.

Acknowledgments

We thank Yijun Feng, one of our colleagues, for providing technical support and insightful discussions of training techniques for pre-trained LAS model. We also thank Odette Scharenborg and Najim Dehak for discussions about the task and the model.

References

[1] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, M. Chen, Z.and Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. Johannes, B. Jiang, C. Ju, B. Jun, P. Legresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, and Raima. Deep speech 2: End-to-end speech recognition in English and Mandarin. In *33rd International Conference on Machine Learning, ICML 2016*, number 33rd International Conference on Machine Learning, ICML 2016, pages 312–321, (1)Baidu Silicon Valley AI Lab, 2016.

[2] M. Bates. The 44 phonemes in English. URL `https://www.dyslexia-reading-well.com/44-phonemes-in-english.html`.

[3] Y. Belinkov and J. Glass. Analyzing hidden representations in end-to-end automatic speech recognition systems. In *Proceedings of Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[4] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)*, pages 4960–4964, May 2016.

[5] J. Garofolo, L. Lamel, W. Fisher, D. P. Jonathan Fiscus, N. Dahlgren, and V. Zue. TIMIT acoustic-phonetic continuous speech corpus LDC93S1, 1993. Web Download.

[6] A. Graves, S. Fernández, and F. Gomez. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning, ICML 2006*, pages 369–376, 2006.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[8] S. Kim, T. Hori, and S. Watanabe. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 4835–4839, 2017.

[9] K.-F. Lee. Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition. *IEEE Trans. on Acoustics, Speech, and Sig. Proc.*, 38, 1990.

[10] J. Lyons. Mel Frequency Cepstral Coefficient (MFCC) tutorial. URL `http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/`.

[11] N. Morgan and H. Bourlard. Continuous speech recognition. In *IEEE Signal Processing Magazine IEEE Signal Process. Mag. Signal Processing Magazine, IEEE.*, volume 12, May 1995.

[12] G. Neubig, C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn, K. Duh, M. Faruqui, C. Gan, D. Garrette, Y. Ji, L. Kong, A. Kuncoro, G. Kumar, C. Malaviya, P. Michel, Y. Oda, M. Richardson, N. Saphra, S. Swayamdipta, and P. Yin. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*, 2017.

[13] G. Neubig, M. Sperber, X. Wang, M. Felix, A. Matthews, S. Padmanabhan, Y. Qi, D. S. Sachan, P. Arthur, P. Godard, J. Hewitt, R. Riad, and L. Wang. XNMT: The extensible neural machine translation toolkit. In *Conference of the Association for Machine Translation in the Americas (AMTA) Open Source Software Showcase*, Boston, March 2018.

[14] J. Novak, P. Dixon, N. Minematsu, K. Hirose, C. Hori, and H. Kashioka. Improving WFST-based G2P conversion with alignment constraints and RNNLM n-best rescoring. In *Interspeech*, 2012.

[15] A. Rousseau, P. Deléglise, and Y. Estève. Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, May 2014.