Interspeech 2022
18-22 September 2022, Incheon, Korea

Cross-lingual articulatory feature information transfer for speech recognition using recurrent progressive neural networks

Mahir Morshed, Mark Hasegawa-Johnson

University of Illinois Urbana-Champaign, Urbana, Illinois 61801
{mmorshe2,jhasegaw}@illinois.edu

Abstract

A system for the lateral transfer of information from end-to-end neural networks recognizing articulatory feature classes to similarly structured networks recognizing phone tokens is here proposed. The system connects recurrent layers of feature detectors pre-trained on a base language to recurrent layers of a phone recognizer for a different target language, this inspired primarily by the progressive neural network scheme. Initial experiments used detectors trained on Bengali speech for four articulatory feature classes—consonant place, consonant manner, vowel height, and vowel backness—attached to phone recognizers for four other Asian languages (Javanese, Nepali, Sinhalese, and Sundanese). While these do not currently suggest consistent performance improvements across different low-resource settings for target languages, irrespective of their genealogic or phonological relatedness to Bengali, they do suggest the need for further trials with different language sets, altered data sources and data configurations, and slightly altered network setups.

Index Terms: articulatory feature detection, transfer learning, progressive neural networks

1. Introduction

Systems that detect individual sub-phone features in speech, such as the placement of vocal tract elements and the airstream mechanism when producing a phone, can be attractive as supplementary sources of information for speech recognition. Just as other linguistic domain knowledge can improve a recognizer's outputs, so too can the incorporation of knowledge about the way individual speech sounds are articulated (and, by extension, the factors that lead to allophony and other consistent sound changes in a given language) improve the overall process of recognizing spoken words. Such knowledge is often useful for target languages without much training data to begin with.

The detection of a particular articulatory feature, or of a specific feature in a feature class, may be considered an individual task to which a model may be fitted. Since identifying a number of distinct articulatory features present in a phone (essentially accomplishing several base tasks) can aid the direct identification of said phone (a target task), the argument may be made that feature detectors trained on one language may aid the general recognition of phones in another. If a given language has a voiced bilabial stop, for example, and there are models that can distinguish each of these feature values in other languages well, then this knowledge about features ideally should be usable in the development of a phone recognizer in that given language.

These are the primary motivations behind the development of the proposed system. It attempts to transfer information between the recurrent layers at equal depths of similarly constructed models, from articulatory feature detectors trained on a "higher-resourced" language to phone recognizers for a "lower-resourced" one, using a scheme inspired by progressive neural networks. The setup does not quite resemble that of a 'teacher-student' relationship, in that the resulting recognizer continues to rely on feature detectors in evaluation, but is meant to encourage the recognizer network to learn only information not otherwise discerned on an articulatory feature basis.

The rest of this paper proceeds as follows. Section 2 provides context on the general tasks of articulatory feature detection and end-to-end transfer learning. Section 3 outlines the architecture used in these experiments and provides information about the datasets for the five languages under consideration. Section 4 presents initial results from the experiments and comments on performance differences between the baselines and the progressive networks. Section 5 provides concluding remarks.

2. Background

2.1. Articulatory feature detection

Systems which rely on articulatory features have taken numerous forms, often similar to those of accompanying networks trained to recognize phones. Some have trained separate multi-class classifiers for different articulatory feature groups, concatenating their outputs with standard mel-frequency cepstral coefficients (MFCCs) as input to a fully-connected network targeting English speech [1]. Others, not primarily relying on recurrent neural networks (RNNs), have trained individual binary articulatory feature detectors using purely fully-connected networks targeting Bengali speech [2]. Yet others have used convolutional neural networks (CNNs) to construct feature detectors for place and manner of articulation [3] targeting Dutch speech, obtaining slight performance increases compared to a multi-layer perceptron (MLP) baseline.

A more recent transformer-based network was developed to identify multiple articulatory features all at once in the sounds of Burmese, Khmer, Nepali, and Sinhalese (both individually and as part of a multilingual recognition system). This network yielded among the lowest character error rates (CERs) in a multilingual setting, particularly when compared to similar systems that identified words, characters, or cross-language-normalized phones [4].

2.2. Transfer learning for end-to-end systems

Pursuing a transfer learning approach when developing an automatic speech recognition (ASR) system provides a better initial state for training such a system on low-resource languages. Not only does this better starting point hasten reaching an optimal state in the network, information about phenomena common to the language of the base system and a new language may be imparted to the new system during training. On this basis a number of different strategies have been employed to transfer

Copyright © 2022 ISCA

10.21437/Interspeech.2022-11202

knowledge between end-to-end systems.

Attempts at fine-tuning non-recurrent, non-attention-based models for speech recognition have often taken considerably different forms. A comparison of a fully-connected network retrained on a new language's speech—either the output layer alone with the target language's data, or the entire network with both languages' data—showed lower CERs with the latter configuration on varied corpora [5]. Sometimes with input and hidden layers (shared across languages) subject to weighting, both while training in the multilingual setting and during specialized adaptation for a new language, slight improvements in word error rate (WER) could be obtained when supplemented with low-rank factorization of hidden layers [6]. After freezing some layers of a wav2letter system, the loss incurred could rival but also still be slightly larger than the loss when retraining an entire network [7].

Similar diversity in the setup of recurrent systems has led to varied outcomes in retraining as well. Experiments suggest that even for very small target language datasets, the CER when retraining all hidden layers of a recognizer on a target language is reduced compared to training the network in a multilingual setting. [8]. Additionally, the training of a multilingual connectionist temporal classification (CTC)-based network on a set of languages through 'fusing' parameters from a network pre-trained on other languages [9] produced mixed WERs improvements across target languages depending on the specific places of fusion chosen—either at the output (weighting the output of the pre-trained model compared to the new network), throughout the new network's hidden layers (using the pre-trained model's parameters as static additions to those in the new network), or only at the gating mechanisms of the layers.

3. Methods

3.1. Architecture

The primary components of the individual feature detectors and phone recognizers here are networks based on the Deep Speech 2 architecture [10]. These networks are thus constructed with two two-dimensional convolutional layers at the input (the "2-layer 2D" configuration from p. 9 of the original paper), followed by five 1024-length long short-term memory (LSTM) blocks, followed by a single fully connected layer generating softmax probabilities for individual features at the output (see the left portion of Figure 1).

The progressive network configuration here [11] treats the detection of individual features as prior tasks, with information obtained (given a particular input) to be transferred to the target task of phone detection (given the same input). All model weights from prior tasks are frozen, while backpropagation within the phone recognition network is otherwise allowed.

Given outputs $h_i^{(f)} \in \mathbb{R}^{1024}$ from the ith recurrent layer of the detector for feature f, the inputs $z_{i+1} \in \mathbb{R}^{1024}$ to layer $i+1$ of the phone recognizer are formed as follows:

$$z_{i+1} = W_{i+1}(h_i^{(\text{phone})} + \sum_{f \in \{\text{place,manner,height,backness}\}} h_i^{(f)})$$

Here the additions are performed element-wise, and $W_i \in \mathbb{R}^{1024 \times 1024}$ acts as a gating mechanism to control how much detector information is propagated to higher recurrent layers of the recognizer network (see the right portion of Figure 1). The choice of a single gating matrix at each layer originates in [12],

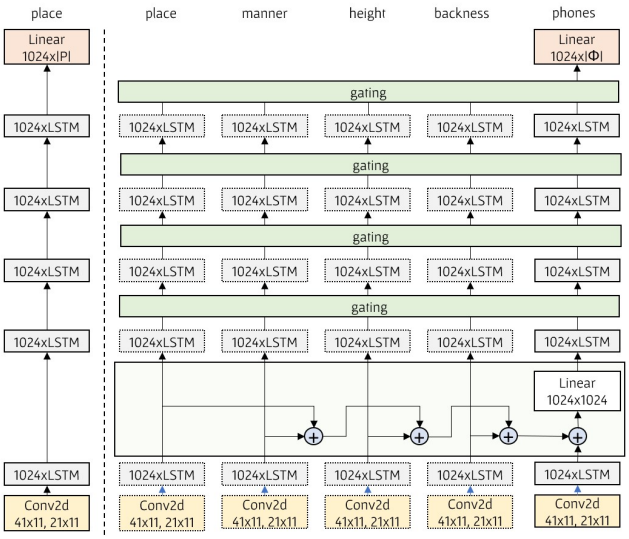

Figure 1: *Diagram of the architectures described. To the left of the dashed line is the feature detector architecture, similar in composition to Deep Speech 2. To the right of the dashed line is the progressive network arrangement of feature detectors and phone recognizer, where the gating layers higher up in the network have the same structure as the expanded layer after the first set of LSTMs. Dotted elements have their weights frozen.*

where a similar progressive network scheme was used to augment end-to-end speech recognizers in a monolingual setting.

3.2. Data

The experiments described here use large speech corpora provided by Google for the Indo-Aryan languages Bengali (`bn`), Nepali (`ne`), and Sinhalese (`si`) and for the Malayo-Polynesian languages Javanese (`jv`) and Sundanese (`su`) [13]. A separate dataset from Google provides, for each of these languages, lexicons with expansions of various text tokens into phone streams and phoneset information [14].

Due to the presence of the aforementioned lexicons, a separate grapheme-to-phoneme (G2P) converter for these languages was not used in preprocessing. (A few entries in these lexicons did not actually have phone information, requiring these to be filled in manually.) Filtering the corpora for those sentences all of whose tokens were in their respective languages' lexicons yielded more than 100h of data for Bengali, Javanese, and Sinhalese, but less than that amount (around 60h) for Nepali and Sundanese. As a result, the Nepali and Sundanese data are treated as purely "low-resource" scenarios in these experiments.

As there are no existing splits of the speech corpora into training, development, and test sets to the best of the authors' knowledge, these were generated for these experiments. In particular, the Bengali, Javanese, and Sinhalese datasets were split into 80h training sets, 10h development sets, and 10h testing sets (hereafter "80h/10h/10h"). For the low-resource settings, the Javanese and Sinhalese datasets were also split into 20h/2.5h/2.5h and 10h/1.25h/1.25h sets, where each of the components of each split is a randomized subset of the corresponding component in the 80h/10h/10h sets of the respective languages. The Nepali and Sinhalese datasets were similarly split into 20h/2.5h/2.5h and 10h/1.25h/1.25h sets, each taking randomized subsets of the corresponding component of

feature class	possible values
place	bilabial, labiodental, dental, alveolar, postalveolar, palatal, velar, glottal
manner	stop, affricate, fricative, nasal, approximant, lateral
height	close, nearclose, closemid, mid, openmid, nearopen, open
backness	front, central, back

Table 1: *Articulatory feature distinctions targeted by each feature detector.*

position	CER	WER
place	5.165	31.969
manner	6.165	33.823
height	5.967	33.076
backness	5.052	33.969

Table 2: *Evaluation results for the Bengali feature detectors after training on the 80h/10h/10h split.*

an 80%/10%/10% split of that language's data. (All of these may be provided on request.)

The primary categories of articulatory feature laid out in this data are place, manner, and voicing for consonants, and height, length, backness, and roundedness for vowels. The components of four of these categories are listed in Table 1; the rest are binary features.

The phoneset information for each of the languages contains mappings of their phone sets to those features for which distinctions are important. All five languages, for example, have mappings of phones to place and manner for consonants and height and backness for vowels. The information for all languages but Bengali also contains mappings for the vowel length binary feature (long or short). The information for Javanese, Nepali, and Sundanese also contains mappings to a binary feature for schwa-like quality, applied to the mid-central vowel in Javanese and Sundanese (as a distinct feature from the others for vowels) and a low front vowel in Sinhalese (as an optional component of the length feature for vowels).

4. Results

The models described in this section were all trained using the CTC loss criterion, with best path decoding for all outputs. Each was trained for 40 epochs with a minibatch size of 20 and a learning rate of $1.5 * 10^{-4}$, annealing this rate by 1.01 after each epoch and using a momentum of 0.9. Inputs to these models were raw spectrograms with a 20ms window size and a 10ms stride (resulting in a 161×100 input per second of 16KHz-sampled audio).

The notion of "character" with respect to CER refers to a single phone, feature token, or space, and that of "word" with respect to WER in these results refers to a sequence of phones/feature tokens delimited by spaces; no transformation of the output into sequences of graphemic units using the lexicons of the target language was performed.

All experiments were performed using the PyTorch deep learning library [15, 16] on an Nvidia GeForce RTX 2080 Ti GPU.

	10h/1.25h/1.25h	20h/2.5h/2.5h	80h/10h/10h
jv	17.483/58.834	12.382/45.722	11.305/38.603
si	(*)	18.830/69.949	11.935/50.127
ne	22.213/74.249	17.034/62.183	n/a
su	3.992/14.552	2.848/9.603	n/a

Table 3: *Evaluation results (*CER/WER*) for the regular phone detectors in the target languages using the test component of the split listed. (*) indicates non-convergence of the model after 40 epochs.*

4.1. Feature detectors

Bengali was chosen as the "high-resource" language for which feature detectors would be trained; this allowed for comparisons of recognizer performance using these detectors with two related languages and two unrelated ones, or alternately with two other "high-resource" and two "low-resource" languages on account of the available data. Articulatory feature detectors were trained with the full 80h/10h/10h split in that language.

The output alphabet for the feature detectors consisted of individual letters representing the members of the feature class, plus either "V" (if a vowel is detected and the feature in question only concerns consonants) or "C" (vice versa), as well as the space character. Feature detectors were trained only on those feature classes common to all languages, partly for consistency in this respect and partly—in the context of excluding all other binary features—in the interest of considering classes for which differences in observed values across languages is possible.

Results for these are shown in Table 2. Although the values reported in [12] for place and manner detectors are not directly comparable to the ones reported here, they at least suggest that articulation information is being acquired well.

4.2. Baselines

Baseline phone recognizers were then trained for each of the splits outlined in the previous section for the four remaining languages. The output alphabet for these was a set of unique Unicode symbols representing different International Phonetic Alphabet (IPA) phones, as well as the space character.

The baselines using the 80h/10h/10h split for Javanese and Sinhalese were initially separately trained 1) to produce tokens from the union of the phone sets for all five dataset languages and 2) to produce tokens from the phone set of the specific language whose data was being used. Because better CERs were achieved for Javanese with the latter phone set (11.305 versus 24.605) and Sinhalese with the former set (11.935 versus 25.505), these were used when training all systems for those respective languages and their genealogical relatives.

Results for all of these baselines are shown in Table 3. The particularly low error rates for Sundanese are most likely due to the data distribution (more on this in the Discussion).

4.3. Progressive networks

Phone recognizers of the same structure as the baseline were trained in the progressive network scheme using all four of the Bengali-trained articulatory feature detectors and the 10h/1.25h/1.25h and 20h/2.5h/2.5h splits. Results for all of these are shown in Table 4.

	10h/1.25h/1.25h	20h/2.5h/2.5h
jv	22.247/69.259	13.608/48.790
si	28.247/85.461	20.110/72.369
ne	(*)	18.138/65.919
su	43.926/59.501	3.169/10.345

Table 4: *Evaluation results (*CER/WER*) for the progressive network-based phone detectors in the target languages using the test component of the split listed. (*) indicates non-convergence of the model after 40 epochs.*

4.4. Discussion

With the sole exception of one language-dataset size pair (since the baseline in that case did not even converge), the progressive network scheme did not yield improvements in performance compared to the baselines; the degradation in performance actually appears more pronounced with smaller training data sizes, notwithstanding the considerably larger error increases with respect to Nepali and Sundanese.

The sources of performance degradation for these setups are likely as numerous as the number of these setups themselves. It is most likely that the two most consequential issues in the cross-lingual setting were 1) the inclusion of the recurrent outputs from the recognizer network in the progressive network gating and 2) areas of non-overlap between the Bengali phone set and the phone sets of the target languages. In particular, 1) may have prevented the proper weighting of focus on individual layer outputs, and 2) may have led to a loss of generality with respect to what the feature detectors identified about their feature classes in training.

The choices of dataset and source of phone transcriptions may have also led to some degradations in the low-resource setting. The overwhelming majority of Sundanese sentences, for example, had the same two overall syntactic forms, such that a model failure with respect to one of those sentence forms would lead to a drastic error rate increase. The average utterance and phone transcription length of Nepali utterances was also much shorter than those for the other four languages. More generally, the phone transcriptions in the provided lexicons may not fully represent the diversity in speakers' pronunciations of word tokens in the respective datasets—this especially on account of the variety in dialects within each of the languages.

5. Conclusions

An attempt at building a model to transfer previously learned information about individual articulatory feature classes to new models for phone recognition has been described. The experiments using this model, however, do not yet support the idea of a performance improvement in the cross-lingual low-resource setting, which may be due to any of various problem sources.

Addressing those issues described in the Discussion would be the most immediate way forward from here. Small architectural changes, such as gating the recurrent outputs of each feature detector layer separately, may be most warranted. Providing data from a set of languages for feature detector training, this data being cleaner and more diverse in content than the datasets used here, may improve the accuracy and generalizability of the detector outputs. More comprehensive measures at tackling the information transfer problem more generally may include introducing binary feature detectors, re-attempting these experiments using attention-based networks, reconstituting these experiments as fine-tuning tasks, and using more linguistically diverse datasets in training (not necessarily entirely overlapping with the languages used here).

6. References

[1] B. Abraham, S. Umesh, and N. M. Joy, "Articulatory feature extraction using CTC to build articulatory classifiers without forced frame alignments for speech recognition," in *Interspeech 2016*, 2016, pp. 798–802.

[2] T. Bhowmik, A. Chowdhury, and S. K. Das Mandal, "Deep neural network based place and manner of articulation detection and classification for bengali continuous speech," *Procedia Computer Science*, vol. 125, pp. 895–901, 2018.

[3] D. Merkx and O. Scharenborg, "Articulatory feature classification using convolutional neural networks," in *Interspeech 2018*. ISCA, 2018, pp. 2142–2146.

[4] S. Li, C. Ding, X. Lu, P. Shen, T. Kawahara, and H. Kawai, "End-to-end articulatory attribute modeling for low-resource multilingual speech recognition," in *Interspeech 2019*. ISCA, 2019, pp. 2145–2149.

[5] C. Lin, Y. Wang, S. Chen, and Y. Liao, "A preliminary study on cross-language knowledge transfer for low-resource taiwanese mandarin ASR," in *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*, 2016, pp. 33–38.

[6] R. Sahraeian and D. Van Compernolle, "Using weighted model averaging in distributed multilingual DNNs to improve low resource ASR," *Procedia Computer Science*, vol. 81, pp. 152–158, 2016.

[7] J. Kunze, L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, and S. Stober, "Transfer learning for speech recognition on a budget," in *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Association for Computational Linguistics, 2017, pp. 168–177.

[8] C. Yu, Y. Chen, Y. Li, M. Kang, S. Xu, and X. Liu, "Cross-language end-to-end speech recognition research based on transfer learning for the low-resource tujia language," *Symmetry*, vol. 11, no. 2, p. 179, 2019.

[9] H. Inaguma, J. Cho, M. K. Baskar, T. Kawahara, and S. Watanabe, "Transfer learning of language-independent end-to-end ASR with language model fusion," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6096–6100.

[10] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. 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, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu, "Deep speech 2 : End-to-end speech recognition in english and mandarin," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 173–182.

[11] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *CoRR*, vol. abs/1606.04671, 2016.

[12] L. Qu, C. Weber, E. Lakomkin, J. Twiefel, and S. Wermter, "Combining articulatory features with end-to-end learning in speech recognition," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, ser. Lecture Notes in Computer Science,

V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds. Springer International Publishing, 2018, pp. 500–510.

[13] O. Kjartansson, S. Sarin, K. Pipatsrisawat, M. Jansche, and L. Ha, "Crowd-sourced speech corpora for javanese, sundanese, sinhala, nepali, and bangladeshi bengali," in *The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*. ISCA, 2018, pp. 52–55.

[14] K. Sodimana, K. Pipatsrisawat, L. Ha, M. Jansche, O. Kjartansson, P. D. Silva, and S. Sarin, "A step-by-step process for building tts voices using open source data and framework for bangla, javanese, khmer, nepali, sinhala, and sundanese," in *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2018, pp. 66–70. [Online]. Available: http://dx.doi.org/10.21437/SLTU.2018-14

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[16] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," in *10.5281/zenodo.3828935*, March 2019.