INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

End-to-End Spoken Language Understanding: Bootstrapping in Low Resource Scenarios

Swapnil Bhosale, Imran Sheikh, Sri Harsha Dumpala, Sunil Kumar Kopparapu

TCS Research and Innovation - Mumbai, India

{swapnil.bhosale2, imran.as, d.harsha, sunilkumar.kopparapu}@tcs.com

Abstract

End-to-end Spoken Language Understanding (SLU) systems, without speech-to-text conversion, are more promising in low resource scenarios. They can be more effective when there is not enough labeled data to train reliable speech recognition and language understanding systems, or where running SLU on edge is preferred over cloud based services. In this paper, we present an approach for bootstrapping end-to-end SLU in low resource scenarios. We show that incorporating layers extracted from pre-trained acoustic models, instead of using the typical Mel filter bank features, lead to better performing SLU models. Moreover, the layers extracted from a model pre-trained on one language perform well even for (a) SLU tasks on a different language and also (b) on utterances from speakers with speech disorder.

Index Terms: SLU, intent classification, low resource.

1. Introduction

Spoken Language Understanding (SLU) deals with the extraction of information and meaning from a speech utterance, in order to facilitate human-machine (and human-human) communication applications such as spoken dialog systems, voice search, topic/domain classification and speech analytics [1]. Traditionally, SLU systems have relied on a pipeline consisting of Automatic Speech Recognition (ASR) module followed by a Natural Language Understanding (NLU) module. The ASR module performs a speech-to-text conversion of the spoken utterance and a task specific NLU module extracts the necessary information from the ASR transcription. While this approach is adopted in most of the existing SLU systems, it is well known that errors in speech-to-text conversion propagate to the NLU module, resulting in errors in SLU. ASR errors can arise from disfluencies in spontaneous utterances, mis-matched domain and noisy conditions [2]. Moreover, performance of SLU degrades for NLU modules trained on manual transcriptions [3] and also when ASR and NLU modules are optimized independently [4].

More recently, researchers have been exploring the idea of end-to-end spoken language understanding [4, 5, 6, 7, 8], without speech-to-text conversion. These approaches take a sequence of spectral/cepstral speech features as input and train a Deep Neural Network (DNN) classifier which can predict the intent and/or slot labels. We believe that this approach is more promising and efficient in scenarios where (a) there is not enough labeled data to train SLU systems [8], (b) general purpose ASR do not perform well for specific group of users, for instance children [9], the elderly [10] and users with speech disorders [11], and (c) running SLU on edge/embedded devices is preferred over cloud based services [12], either due to privacy concerns [13] or network bottleneck or computation constraints.

This paper presents an approach for bootstrapping end-to-end SLU systems. Previous work [4, 6, 7] have shown that SLU systems without speech-to-text conversion can achieve performance comparable to SLU systems with ASR+NLU pipeline. However, this performance was achieved when systems were trained on millions of labeled speech utterances. End-to-end SLU has been shown to work in a low resource speaker dependent scenario [8], where DNN models were trained and tested on the same speaker. As expected, this approach leads to severe degradation in SLU performance in a speaker independent scenario, as shown in [5] and also validated by experiments presented in this paper. We hypothesize that these DNN models are unable to deal with speaker variability when trained with small amounts of speech data, for classifying intents and slots in the SLU task. In this paper, we show that an end-to-end SLU systems can be bootstrapped by reusing layers from pre-trained ASR acoustic models.

Transfer learning approaches have been applied to speech processing tasks in different settings [14]. They have been typically explored for re-using ASR models for different domains [15], multiple languages [16], children speech recognition [17] as well as for speaker and spoken language recognition tasks [18]. This paper explores a transfer learning approach to bootstrap end-to-end SLU in low resource task. We explore layers extracted from popular DNN architectures for ASR, namely an end-to-end trained DeepSpeech model [19] and a Time Delay Neural Network (TDNN) based ASR acoustic model [20]. We show that training intent and slot classification models with these pre-trained layers can lead to usable speaker independent SLU systems in low resource scenarios. We further validate our approach by (a) training the SLU model in one language with the pre-trained layers obtained from a model trained on a different language and (b) training the SLU model on disordered speech (along with a language mismatch scenario).

The rest of the paper is organized as follows. Section 2 presents our end-to-end SLU architecture and an analysis of different layers from pre-trained ASR models. Section 3 details the experiments and results which validate our proposed approach, followed by a discussion on the learned models in Section 4 and the conclusion in Section 5.

2. End-to-End SLU

Traditional SLU systems process an input spoken utterance, represented as a sequence of frame-level features $X = \{x_1, x_2, ..., x_T\}$, using an ASR $f()$ to obtain a sequence of words; namely,

$$W = f(X) = f(x_1, x_2, ..., x_T) \quad (1)$$

where $W = w_1, w_2, ..., w_N$. This word sequence is processed by an NLU module $g()$ to extract a set of intent and slot labels:

$$S = g(W) = g(w_1, w_2, ..., w_N) \quad (2)$$

http://dx.doi.org/10.21437/Interspeech.2019-2366

where $S = \{s_1, s_2, ..., s_M\}$. An end-to-end SLU without speech-to-text, on the other hand tries to achieve the same task of extracting intent and slot labels by directly processing the speech sequence, namely

$$S = h(X) = h(x_1, x_2, ..., x_T) \quad (3)$$

Typically, $h()$ can be modeled using an encoder-decoder architecture [5, 6, 4, 8]. The encoder processes the variable length speech sequence (X) to obtain an encoded vector representation of the utterance and the decoder classifies intent and slot labels from this encoded vector.

2.1. Bootstrapping in Low Resource Scenario

Previous work on SLU without speech-to-text conversion have used Mel Filter Bank (MFB) output features [4, 6, 8] or the Mel Frequency Cepstral Coefficients (MFCCs) [5] to represent the input speech utterance. Encoder-decoder models trained on these features are data hungry and as a result show severe degradation in SLU performance in low resource speaker independent scenarios [5]. We hypothesize that, with small amounts of training speech, the SLU model is unable to handle speaker variability and learn the temporal context simultaneously. It should be noticed that though there exist techniques for speaker normalization and handling speaker variability in the ASR literature [21, 22], we choose to explore and exploit the representations learned by DNN based ASR models.

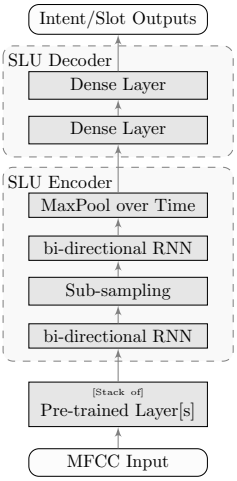

Figure 1: *Model for bootstrapping end-to-end SLU.*

Figure 1 shows the architecture of our model bootstrapping end-to-end SLU in low resource scenarios. In our model, the sequence of input MFCC features (X) is first passed through a set of pre-trained layers. This is followed by the encoder-decoder SLU model, similar to [6], wherein the encoder consists of two bi-directional Recurrent Neural Network (RNN) layers with a sub-sampling layer in between the two RNN layers and a max-pooling over T timesteps to obtain the encoding vector. The decoder consists of two fully connected dense layers with ReLU activation in between them. Sigmoid activations are used in the final output layer in order to achieve a multi-label multi-class classification over the (M) possible intent and slot labels.

To identify the set of pre-trained layers which are most suitable for the task, we explore two publicly available pre-trained models i.e., DeepSpeech [19, 23] and ASpIRE ASR [20, 24] DNN models.

2.1.1. Pre-trained layers from DeepSpeech model

The DeepSpeech model consists of three fully connected dense layers followed by a bi-directional RNN layer and then two fully connected dense layers [19]. The final layer outputs scores over the possible output characters. We specifically use the Mozilla DeepSpeech model [23] which is pre-trained on 1000 hours of English Librispeech data. As compared to [19], this Mozilla DeepSpeech model uses only a forward direction RNN layer with Long-Short-Term Memory (LSTM) units. The input utterance is represented by a sequence of frames, with each frame comprising 26-dimensional MFCCs and additionally a context of 18 adjacent frames (9 preceding + 9 following frames).

Each input frame (along with its context) will be transformed into an intermediate representation by each of the layers of the DeepSpeech model. While the activations of the final layer representing output character labels might seem to be most useful for our task, they might not work well for training SLU models for languages other than English. Moreover, we hypothesize that speaker invariance and temporal context might be already captured in the earlier layers of the model. In order to choose from the available intermediate layer representations, we visualize the representations from all the layers of the DeepSpeech model and analyze the structure and spread of these representations.

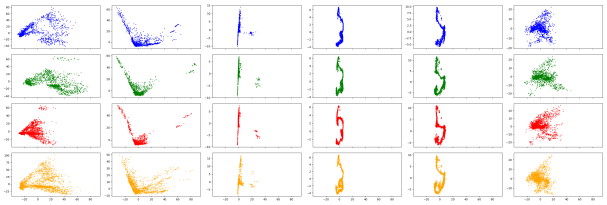

Figure 2: *Visualization of activations from different layers of DeepSpeech model, for a Dutch utterance spoken by different native speakers. Each row (with different colors) represents a speaker. Columns represent the 6 layers of the model, wherein fourth layer is LSTM and remaining are dense.*

Figure 2 shows the 2-dimensional PCA plots of the outputs, corresponding to each input frame (along with its context), from different layers of the DeepSpeech model. It can be observed that the output of the first two dense layers of the DeepSpeech model (first two columns) vary across the rows i.e., different speakers. However, the other layer outputs (columns $3, 4, 5, 6$; of Figure 2) seem to have similar pattern across different speakers (rows), especially the LSTM layer output (column 4) and the layers around the LSTM layer (columns 3 and 5). In our experiments, we slice the DeepSpeech model at these layers and use it as the pre-trained layers in our SLU model.

2.1.2. Pre-trained layers from ASpIRE Acoustic model

The ASpIRE acoustic model [20] consists of a stack of TDNN layers, with different input contexts, and a sub-sampling which selects non-contiguous temporal frames, at each layer. We use the publicly available ASpIRE acoustic model [24] which is pre-trained on the Fisher English corpus. It consists of 6 layers of TDNNs, each with ReLU activations and re-normalization, followed by two branches; one for sequence (Chain) training and the other for cross-entropy (Xent) regularization. The input utterance is represented by a sequence of frames, with each frame comprising 40-dimensional MFCCs and a 100-

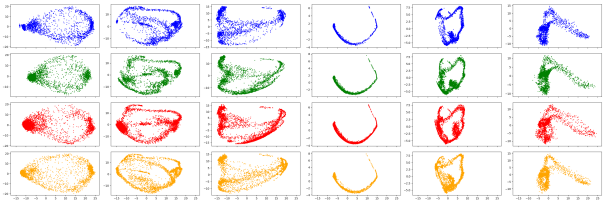

Figure 3: *Visualization of activations from different layers of ASpIRE model, for a Dutch utterance spoken by multiple speakers. Each row (with different colors) represents a speaker. Columns represent layers TDNN0, TDNN4, TDNN5, pre-final Chain layer, pre-final Xent layer and output,respectively.*

dimensional i-vector feature. The output of the final layer represents the scores across 8629 context-dependent phone states (senones). Figure 3 shows visualizations of outputs from different layers of the ASpIRE acoustic model. In case of the ASpIRE model, consistent patterns across speakers are seen from TDNN layer 4 and beyond. In our experiments, we splice the model until layer 3, and beyond, to form the pre-trained layers. We skip the final outputs because of their high dimensionality, which leads to too many parameters in the SLU model.

3. Experiments

In order to evaluate the proposed methodology for bootstrapping SLU systems, we conducted experiments on three different SLU setups (1) SmartLights English assistant: meant to control lights in a room [12], (2) Grabo: for controlling movements of a robot in Dutch [25], and (3) Domotica-4: Dutch dysarthric speakers using spoken utterances in a home automation task [26]. Details of these tasks, the experiment setup and the results obtained are presented in this section. The baseline system is an encoder-decoder SLU model with MFB features as inputs, which performed better than MFCC features, and without any pre-trained layers. The baseline system is compared to the proposed SLU model with pre-trained layers and the encoder-decoder, as discussed in Section 2.1.

Table 1: *Dataset Description. ('I, S' denote intents and slots.)*

Dataset	Language	#speakers	#utt	#I, #S
SmartLights	English	52	1660	6, -
Grabo	Dutch	10	6350	8, 32
Domotica-4	Dutch	7	1164	4, 22

3.1. Datasets and Experimental Setup

The SmartLights assistant dataset [12] consists of spoken utterances comprising 6 intents allowing to turn on or off the light or to change its brightness or color. Each utterance can have one or more slots filled by the name of the room and/or the color/brightness of light. However, we only consider the intent classification task because many slot values occur infrequently in the dataset. As shown in Table 1, the original dataset consists of 1660 utterances spoken by 52 speakers. In our experiments, we split the dataset in the ratio 70:10:20, representing train, validation and test sets, such that no speaker is spread across these sets. The SmartLights dataset validates the proposed approach for speaker independence, in a scenario where the language of the SLU task is same as that of the model from which pre-trained layers are extracted.

The Grabo dataset [25] consists of Dutch spoken utterances controlling movements of a robot. As shown in Table 1, it consists of 6350 utterances spoken by 10 speakers, where each utterance can have one of the possible 8 intents representing the movement and one or more of the 32 slots representing the position of the robot. Since this dataset has only 10 speakers, we perform a 10-fold cross validation wherein all utterances belonging to a single speaker form the test set. In each fold, data from 1 out of the remaining 9 speakers is used as a validation set and the rest of the data is used as train set. Each fold is repeated 3 times to choose a random set of validation and train speakers. This dataset validates our approach in a scenario where the language of the SLU task is different from that of the model from which pre-trained layers are extracted.

The Domotica-4 dataset is an extension of the Domotica-3 and Domotica-2 datasets [26]. It consists of recordings from Dutch dysarthric speakers using spoken utterances in a home automation task. As shown in Table 1, it consists of 1164 utterances from 7 dysarthric speakers. Each utterance contains one of the 4 intents and any of the 22 slots which together represent tasks like opening/closing of door or switching on/off light in specific rooms. For this dataset, we perform a 7-fold cross validation similar to that for the Grabo dataset. The Domotica-4 dataset validates our bootstrapping approach on a mis-matched language (similar to Grabo), additionally on speakers with speech disorder.

3.2. SLU Model Training Details

An SLU model is trained for each of the SLU task described in Section 3.1. The bi-directional RNNs in the encoder consists of 256 Gated Recurrent Units (GRU) units. Sub-sampling the number of time steps by 2 gives a better performance. Increasing the number of RNN layers or the sub-sampling hyper-parameter did not help. The size of the fully connected dense layers in the decoder are adjusted as per the size of the encoder representation and the number of output labels (i.e. intents+slots) in the task. We use an ADAM optimizer to train the model. Training proceeds until an early stopping occurs based on the performance on the validation set. Updating the pre-trained layers, with the small amount of speech data available for training, did not result into better performing models.

3.3. Results

Table 2 presents the performance of the different SLU models for different tasks. Performance on the SmartLights task is evaluated in terms of intent classification accuracy whereas the performance on the Grabo and Domotica-4 datasets is evaluated in terms of F1 score, since the task is that of multi-label multi-class classification over intents and slots. F1 scores and accuracy are presented for MFB baseline without pre-trained layers, SLU model with pre-trained layers from DeepSpeech model and SLU model with pre-trained layers from ASpIRE acoustic model.

As seen in Table 2, the use of pre-trained layers (DeepSpeech and ASpIRE) give better performance than the MFB features. While the use of MFB features was shown to perform well in [8] in a speaker dependent setup on the Grabo and Domotica datasets, they fail to perform in a speaker independent train-test setup in low resource scenarios. Secondly, we can observe that the DeepSpeech model, compared to ASpIRE, gives the best performance across all the three tasks. More specifically, representations from the L4 of the DeepSpeech model, which is the LSTM layer, gives the best performance (except

Table 2: *Performance of different pre-trained layers on SLU tasks. ('DS' denotes DeepSpeech, 'L' denotes layer and 'PF' denotes pre-final layer.)*

Input / Pre-trained layer	SmartLights (Accuracy)	Grabo (F1)	Domotica-4 (F1)
MFB (Baseline)	0.4796	0.7018	0.5421
DS-L3	0.6363	0.8344	0.6638
DS-L4 (LSTM)	0.8338	**0.9181**	**0.7904**
DS-L5	0.8369	0.9097	0.7238
DS-Output	**0.8432**	0.8959	0.6847
ASpIRE-TDNN3	0.7492	0.7866	0.6990
ASpIRE-TDNN4	0.7805	0.8718	0.6832
ASpIRE-TDNN5	0.7335	0.8608	0.7028
ASpIRE-Chain-PF	0.7554	0.8400	0.4414
ASpIRE-Xent-PF	0.8181	0.8804	0.7083

for the SmartLights task where it performs slightly lower than the DeepSpeech output).

The DeepSpeech output activations, across English character labels, give the best performance (0.8432) on the SmartLights task. However, the representations from DeepSpeech layers L4 and L5 give a similar performance in this task. DeepSpeech output activations also perform well for the Grabo dataset but it is lower (by 2.2% absolute) than the DeepSpeech L4 LSTM output activations. This can attributed to the mismatch in languages of the DeepSpeech model (English) and that of Grabo (Dutch). In case of the dysarthric Dutch speakers of the Domotica-4 dataset, the DeepSpeech LSTM layer activations outperform the other representations.

Comparison of pre-trained layers from the ASpIRE acoustic model shows that the pre-final Xent layer gives the best performance. The pre-final chain layer, which is trained for speech-to-text conversion with a sequence training objective function, does not perform as well as the pre-final Xent layer. Interestingly, the TDNN4 layer performs better than the pre-final chain layer across all the tasks, and even the TDNN5 layer except for the SmartLights task.

4. Discussion

4.1. Evaluation of Task Completion

Table 2 presents the performance of the SLU tasks in terms of F1 scores but it does not give a clear evaluation in terms of completion of the task, wherein the intent as well as all the slots are perfectly parsed. Table 3 presents the task completion accuracy of the different SLU models on the Grabo and Domotica-4 datasets. We can observe from Table 3 that the DeepSpeech LSTM layer activations perform significantly better than the other systems and that they clearly outperform the MFB features in low resource scenarios.

4.2. Comparison of Saliency Maps

In order to understand the differences in the encoder-decoder SLU models trained with the MFB features and the DeepSpeech-L4 representations, we analyze saliency maps [27] obtained from these two models. The saliency maps that we use represent the gradients computed by calculating the derivative of the model output with respect to the model inputs, keeping the model parameters fixed to those obtained during training.

Figure 4 presents a visualization of MFB features of an utterance from the SmartLights test set, with intent label

Table 3: *Task completion accuracy for different pre-trained layers . ('DS' denotes DeepSpeech, 'L' denotes layer and 'PF' denotes pre-final layer.)*

Input / Pre-trained layer	Grabo (Task Accuracy)	Domotica-4 (Task Accuracy)
MFB (Baseline)	0.6623	0.2642
DS-L3	0.7651	0.4608
DS-L4 (LSTM)	**0.8760**	**0.6306**
DS-L5	0.8615	0.5362
DS-Output	0.8286	0.4716
ASpIRE-TDNN3	0.7398	0.5224
ASpIRE-TDNN4	0.8039	0.4832
ASpIRE-TDNN5	0.7942	0.5426
ASpIRE-Chain-PF	0.7599	0.2573
ASpIRE-Xent-PF	0.8039	0.5356

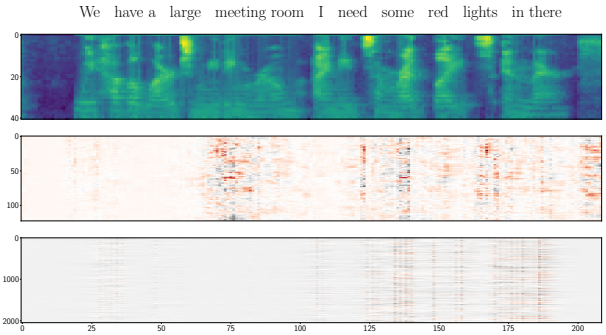

Figure 4: *Visualization of Mel filter bank features of an utterance (top) and, saliency maps obtained from encoder-decoder SLU models with MFB features as input (middle) and DeepSpeech-L4 representations as input (bottom), respectively. (The utterance belongs to the* 'SWITCHLIGHTON' *intent.)*

'SWITCHLIGHTON', and its corresponding saliency maps. The input gradients in Figure 4 show that the encoder-decoder SLU model with DeepSpeech-L4 representations focuses at the relevant locations (/need some red lights/) to classify the SWITCHLIGHTON intent. However, the input gradients for the SLU model with MFB features are spread even in the regions not relevant for classifying the SWITCHLIGHTON intent. The SLU model with DeepSpeech-L4 representations is able to learn this from the small amount of training data because the pre-trained DeepSpeech model has already learned to model temporal context in a sequence of speech frames as well to normalize speaker variability. Hence, it helps to bootstrap a working SLU model.

5. Conclusion

End-to-end SLU systems trained on Mel filter bank based features fail to perform in a speaker independent low resource scenario. Layers extracted from pre-trained ASR models can significantly boost the performance of these SLU models and help to bootstrap end-to-end SLU without speech-to-text conversion in low resource scenarios. Results from our experiments demonstrate that the proposed approach can be used for bootstrapping an SLU system in a different target language as well as systems usable by speakers with speech disorder. Furthermore, saliency map analysis reveals that SLU models with pre-trained layers are better at attending to the relevant regions for the language understanding task.

6. References

[1] G. Tur and R. De Mori, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. Wiley, 2011.

[2] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5934–5938.

[3] E. Simonnet, S. Ghannay, N. Camelin, and Y. Estève, "Simulating asr errors for training slu systems," in *Proceedings of the 11th Language Resources and Evaluation Conference*, May 2018, pp. 3157–3162.

[4] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2018, pp. 720–726.

[5] Y. Qian, R. Ubale, V. Ramanaryanan, P. Lange, D. Suendermann-Oeft, K. Evanini, and E. Tsuprun, "Exploring asr-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017, pp. 569–576.

[6] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5754–5758.

[7] Y. Chen, R. Price, and S. Bangalore, "Spoken language understanding without speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 6189–6193.

[8] V. Renkens and H. Van hamme, "Capsule networks for low resource spoken language understanding," in *Interspeech*, September 2018, pp. 601–605.

[9] H. Liao, G. Pundak, O. Siohan, M. Carroll, N. Coccaro, Q.-M. Jiang, T. N. Sainath, A. Senior, F. Beaufays, and M. Bacchiani, "Large vocabulary automatic speech recognition for children," in *Interspeech*, September 2015, pp. 1611–1615.

[10] F. Aman, M. Vacher, S. Rossato, and F. Portet, "Analyzing the performance of automatic speech recognition for ageing voice: Does it correlate with dependency level?" in *Proceedings of the Fourth Workshop on Speech and Language Processing for Assistive Technologies*, 2013, pp. 9–15.

[11] F. Rudzicz, "Using articulatory likelihoods in the recognition of dysarthric speech," *Speech Communication*, vol. 54, no. 3, pp. 430 – 444, 2012.

[12] A. Saade, A. Coucke, A. Caulier, J. Dureau, A. Ball, T. Bluche, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, and M. Primet, "Spoken language understanding on the edge," *CoRR*, vol. abs/1810.12735, 2018.

[13] J. Lau, B. Zimmerman, and F. Schaub, "Alexa, are you listening?: Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, pp. 102:1–102:31, Nov. 2018.

[14] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Dec 2015, pp. 1225–1237.

[15] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Investigation of transfer learning for asr using lf-mmi trained neural networks," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017, pp. 279–286.

[16] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafit, S. Watanabe, and T. Hori, "Multilingual sequence-to-sequence speech recognition: Architecture, transfer learning, and language modeling," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2018, pp. 521–527.

[17] M. Matassoni, R. Gretter, D. Falavigna, and D. Giuliani, "Non-native children speech recognition through transfer learning," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6229–6233, 2018.

[18] M. McLaren, L. Ferrer, and A. Lawson, "Exploring the role of phonetic bottleneck features for speaker and language recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5575–5579.

[19] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014.

[20] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, "Jhu aspire system: Robust lvcsr with tdnns, ivector adaptation and rnn-lms," in *2015 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015, pp. 539–546.

[21] D. Y. Kim, S. Umesh, M. J. F. Gales, T. Hain, and P. C. Woodland, "Using vtln for broadcast news transcription," in *INTERSPEECH*, 2004, pp. 1953–1956.

[22] S. P. Rath, D. Povey, K. Veselý, and J. Cernocký, "Improved feature processing for deep neural networks," in *INTERSPEECH*, August 2013, pp. 109–113.

[23] Mozilla, "Deep speech 0.4.0," https://github.com/mozilla/DeepSpeech/releases, Jan 2019.

[24] D. Povey, "Aspire chain model," http://kaldi-asr.org/models/m1, Oct 2016.

[25] V. Renkens, S. Janssens, B. Ons, J. F. Gemmeke, and H. Van hamme, "Acquisition of ordinal words using weakly supervised nmf," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2014, pp. 30–35.

[26] B. Ons, J. F. Gemmeke, and H. V. hamme, "The self-taught vocal interface," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 1, p. 43, Dec 2014.

[27] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2013.