Phoneme based Domain Prediction for Language Model Adaptation

Anmol Bhasin
Voice Intelligence R&D
Samsung R&D Institute
Bangalore, India
anmol.bhasin@samsung.com

Gaurav Mathur
Voice Intelligence R&D Samsung
R&D Institute
Bangalore, India
gaurav.m4@samsung.com

Promod Yenigalla
Voice Intelligence R&D Samsung
R&D Institute
Bangalore, India
promod.y@samsung.com

Bharatram Natarajan
Voice Intelligence R&D Samsung
R&D Institute
Bangalore, India
bharatram.n@samsung.com

***Abstract*— Automatic Speech Recognizer (ASR) and Natural Language Understanding (NLU) are the two key components for any voice assistant. ASR converts the input audio signal to text using acoustic model (AM), language model (LM) and Decoder. NLU further processes this text for sub-tasks like predicting domain, intent and slots. Since input to NLU is text, any error in ASR module will propagate in NLU sub-tasks. ASR generally process speech in small duration windows and first generates phonemes using Acoustic Model (AM) and then Word Lattices using Decoder, Dictionary and Language Model (LM). Training and maintaining a generic LM, which fits the distribution of data of multiple domains is a difficult task. So our proposed architecture uses multiple domain specific LMs to rescore word lattice and has a way to select LMs for rescoring. In this paper, we are proposing a novel Multistage CNN architecture to classify the domain from partial phoneme sequence and use it to select top K domain LMs. The accuracy of multistage classification model based on phoneme input for top three domains has achieved state-of-the-art results on 2 open datasets, 97.76% in ATIS and 99.57% in Snips.**

Keywords—Language Adaptation, Phoneme Classification, Multistage CNN, Domain specific LM.

I. INTRODUCTION

Voice based interaction with smart devices is becoming popular for example Chatbot applications and personal assistants. Recent developments, such as Samsung's Bixby, Apple's Siri, Amazon's Alexa and many more are helping this seamless interaction to meet the user expectation. These voice based interactive systems usually work in two steps: conversion of input voice to text using ASR and extracting the information from text using NLU to perform intended action.

A generic Kaldi [1] based ASR system block diagram is shown in Figure 1. For given speech, ASR generates most likely word sequence. Raw audio is processed synchronously on small duration windows (frames) using STFT (Short-Term Fourier Transform) and then acoustic features are generated. Mostly used acoustic features are Mel Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Prediction (PLP). Acoustic model get trained on acoustic features (MFCC or PLP) and predicts senones. Phoneme sequence is generated from these senones. A generic LM provides probability of a word sequence.

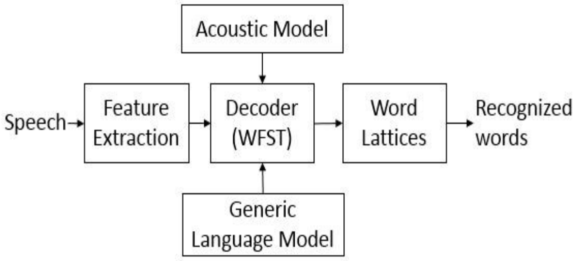

Fig. 1. General Kaldi Based ASR System

Decoder creates Weighted Finite-State Transducers (WFST) Decoding Graph, which takes into account the grammar of data, as well as the distribution and probabilities of contiguous specific words. We can train the WFST to transform the AM output into the desired lattices.

ASR performance depends on many factors like accented speech, pronunciation variation, homophones, speaker variability. Generally, there are post-processing modules to correct ASR errors like homophones correction, LM correction etc. A good example for homophones can be a 'mine' and 'nine'. Based upon the context hearing "I would like mine bags," does not make sense, so the person said either "my" or "nine". In a stationery shop, "nine" makes sense but at airport customs, "my" makes more. Table I shows example of some spoken utterances with homophones. Column 1 of Table I represents utterances that are having some error due homophones, and column 2 of same table shows corresponding correct utterance.

TABLE I. TABLE SHOWING EXAMPLE OF SOME UTTERANCES WITH HOMOPHONES

Wrong Utterance Prediction	Correct utterance Prediction
Other This Week	Weather This Week
Log The Screen After Thirty Seconds	Lock The Screen After Thirty Seconds
Open My Fun	Open My Phone
Calls Google Music	Close Google Music
Sync Me A Festive Rap	Sing Me A Festive Rap

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

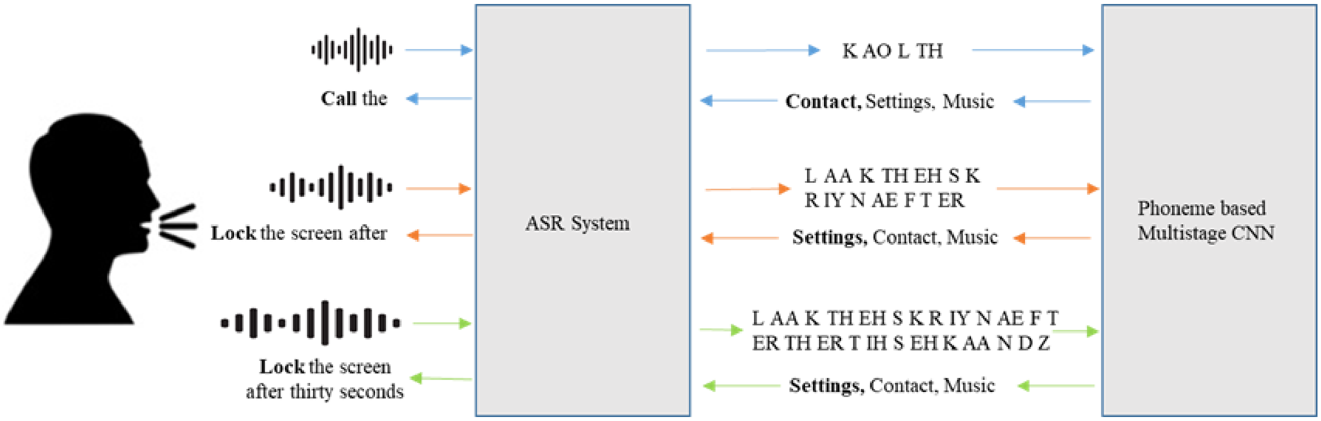
Call the
Lock the screen after
Lock the screen after thirty seconds
ASR System
K AO L TH
Contact, Settings, Music
L AA K TH EH S K R IY N AE F T ER
Settings, Contact, Music
L AA K TH EH S K R IY N AE F T ER TH ER T IH S EH K AA N D Z
Settings, Contact, Music
Phoneme based Multistage CNN

Fig. 2. Diagram showing the working of proposed architecture.

Since the text generated from phoneme sequence depend upon LM and maintaining a generic LM for all the domains over time is difficult task. We propose to maintain domain specific LMs and rescore word lattice based on one or multiple domain specific language models. In this paper our main focus is to present phoneme based multi-stage classifier which can be used to process partial phoneme sequence for selecting domain specific LMs to correct the text generated by ASR. We have chosen phoneme sequences instead of word sequence to mitigate the issue of homophones, where similar phoneme sequence would result in multiple similar sounding words. As AM is generating senones, we are first converting predicted top senones to phoneme sequence and using these phonemes for classification.

Figure 2 shows the end-2-end working of our proposed architecture for utterance 'Lock the screen after thirty seconds'. The expected domains are related to mobile phone applications like contact, messages, settings, clock, calendar etc. The input signal is passed to ASR system, which converts voice to phonemes. These phoneme are used to predict domain like in case of first time frame generated partial phoneme 'K AO L TH' is used to predict top 3 domains (Contact, Settings, Music) and ASR generated text as 'Call the' using Contact Domain LM. In next time frame where input size is increased 'settings' domain is predicted. Now ASR corrects its output and generates 'Lock the screen after'. Similarly at last 'settings' domain is again predicted and complete utterance 'Lock the screen after thirty seconds' is generated. It is not necessary to use only top predicted domain, top k domains can be used for ASR n-best results.

A. Literature study:

Though voice based action identification for interacting with devices has been an area of research since last decade [2-5], a lot of work has been done in improving errors in ASR module [6-12].

In [6] authors tried to auto correct out of words vocabulary prediction not included in LM by using a phrase-based machine translation system trained on words and phonetic encoding representations from n-best lists of ASR results. Luis Fernando et al. [7] used Bing Spelling suggestion for post error correction in predicting the text from ASR. Recently [8] used RNN Language Models for improving ASR error detection rate. In [9] authors categorized system into two stages error detection and error type classification by fetching the generic features obtained from recognizer output and using variant RNN based models on top of them for error detection and classification. To tackle ASR Error outputs Errattahi Rahhal et al. [10] used a classifier-based approach for both error detection and classification by handling recognizing errors separately from ASR decoder. Recently Shivakumar et al. [11] used neural network considering long-term context for improving ASR output possibilities and handling unseen word.[13-14] showed the benefit of using partial utterance that can be used as a feedback, but these are based on text partial utterance as input.

Inspired by all these prior work we designed phoneme based partial input classifier that gives domain prediction that will help in choosing LM for the ASR text generation. The detail method is explained in the following section.

II. PROPOSED METHOD

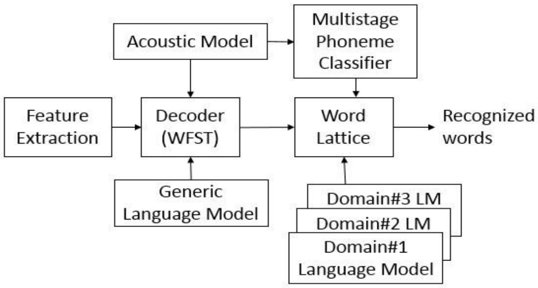

Fig. 3. Proposed ASR System.

Figure 3 shows our proposed ASR system. We proposed an architecture for domain classification using partial phoneme sequence to give top-k domains to ASR, which will aid in choosing Domain LM for rescoring word lattice. This decreases errors due homophone in text conversion. Phoneme based classifier has multistage CNN architecture, which can be used to process partial ASR output at every stage. The detail architecture of the proposed classification method is discussed in the following section.

A. *Phoneme Embedding*

As shown in Table II, the phoneme embedding we generated from word2vec[15] consist of 39 phonemes. In phoneme embedding each phoneme was represented using vector of dimension of 300. Phoneme embedding just like word embedding captures the contextual relationship between various phonemes that appear in a sequence. Therefore even though similar phonemes appear in a sequence the context, which is the real intent of the user, helps in identifying the actual domain.

TABLE II. PHONEME SET.

Phonemes
AA, AE, AH, AO, AW, AY, B, CH, D , DH, EH, ER, EY, F, G, HH, IH, IY, JH, K, L, M, N , NG, W, OY, P, R, S, SH, T, TH, UH, UW, V, W,Y ,Z , ZH

B. *Multistage CNN Architecture using Phoneme as input (Model 1)*

The high-level block diagram of proposed classification system is shown in Figure 4. To generate phoneme sequence for training, we process ATIS and Snips datasets (mentioned in Table III), using g2p-seq2seq [16] model similar to [17], where authors used phonemes for emotion detection. Then using these generated phonemes and Word2Vec [15] we created phoneme embedding of size 40*300. Where 300 is the size of embedding vector and 40 is the number of unique phonemes (39 phonemes and also considering space as a randomly initialized vector of size 300).

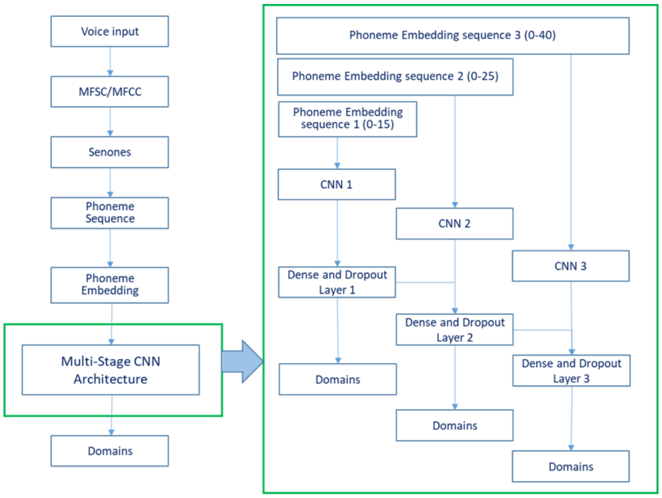
Voice input
MFSC/MFCC
Senones
Phoneme Sequence
Phoneme Embedding
Multi-Stage CNN Architecture
Domains
Phoneme Embedding sequence 3 (0-40)
Phoneme Embedding sequence 2 (0-25)
Phoneme Embedding sequence 1 (0-15)
CNN 1
CNN 2
CNN 3
Dense and Dropout Layer 1
Dense and Dropout Layer 2
Dense and Dropout Layer 3
Domains
Domains
Domains

Fig. 4. High-level block diagram of Phoneme based Classifier.

For our experiments we used three stages. The number of stages can be increased or decreased based on dataset distribution. For first stage, we gave input phonemes of length 15 (0-15), for second stage phonemes of length 25 (0-25) and for final stage phonemes of length 40 (0-40). While giving the input to each stage we ensure that whole word phoneme representation is passed. In case of incomplete phoneme length, padding was applied. Intention behind choosing phonemes of length 15 and 25 is to replicate the scenario where ASR will be sending streaming phonemes in time frame like 200ms. In last stage we gave 40 phonemes as these corresponds to 7-8 words, on average, and should be sufficient to predict correct domain.

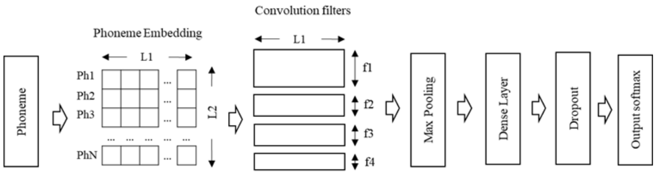

Fig. 5. CNN Module of each Stage.

Figure 5. shows CNN Module used in each stage. The CNN model at each stage was a parallel net similar to [18] containing four parallel CNN layers. The input shape of each CNN was L_1*L_2 where L_1 was size of phoneme embedding i.e. 300 and L_2 is phoneme sequence of length 15, 25 and 40 for three stages respectively.

The convolution filters applied were of size f_1*L_1, f_2*L_1, f_3*L_1 and f_4*L_1, where $f_1=1$, $f_2=2$, $f_3=3$ and $f_4=5$. The number of filters of each type was taken as 128. All the features from convolution filters were concatenated and max polling was applied on it. It was then passed to dense layer of size 256. Output of dense layer was feed to dropout layer with dropout rate as 0.5. The output of dropout was passed by Softmax Layer to predict the label.

To establish relationship between multiple stages of the proposed architecture we concatenated the output of previous stage dense layer with next stage dense layer. This helped to use already learnt information from streaming ASR for next time frames in next stage. The loss from all the 3 stages were summed and used for back propagation.

The primary reason for choosing a multistage CNN model, was to process the streaming voice input in shorter time frames (like 200 milliseconds) and predict the output domain on the partial input which will help to choose domain specific LM in real world end-to-end system as shown in Figure 2.

C. *Multistage CNN Architecture using Phoneme as input and Shared Parameters (Model 2)*

In Model 1 since we are using three stages and each stage is similar to Figure 5, this leads to increase in number of parameters. So, in Model 2, we tried to reduce number of parameters by using same convolution layer and dense layer across three stages instead of re-initialization. Sharing of parameters stabilizes training and helps in model generalization. It also reduces the model size significantly.

III. EXPERIMENTS & DATASETS

We consider two open source datasets 'ATIS' and 'Snips' to evaluate on proposed architecture. All datasets were taken from the GitHub Source mentioned with Table III.

- **ATIS Dataset:** The ATIS (Airline Travel Information Systems) dataset consists of user-spoken utterances for flight reservation. It consist of 4,978 train utterances, 893 utterances as test data and 500 as validation data. The total number of unique labels to be predicted is 21.
- **Snips Dataset:** Snips dataset is collected using Snips personal voice assistant. In Snips data for each intent is uniformly distributed. Train set consists of 13,084 utterances; validation and test both have 700 utterances each. Number of unique labels present are 7.

TABLE III. DETAILS OF ATIS AND SNIPS DATASET USED IN EXPERIMENT

Dataset	ATIS[a]	Snips[b]
Train Data	4,978	13,084
Test Data	893	700
Validation Data	500	700
Vocabulary Size	722	11,241
Unique Labels	21	7

[a] https://github.com/yvchen/JointSLU/tree/master/data

[b] https://github.com/MiuLab/SlotGated-SLU/

TABLE IV. SOME EXAMPLE UTTERANCES AND DOMAIN FROM ATIS AND SNIPS AND THERE PHONEME CONVERSION

Text Sequence	Phoneme Sequence	Label
what flights are available from pittsburgh to baltimore on Thursday morning	W AH T F L AY T S AA R AH V EY L AH B AH L F R AA M P IH T S B ER G T UW B AO L T AH M AO R AO N TH ER Z D IY M AO R N IH NG	atis_flight
what kind of ground transportation is available in denver	W AH T K AY N D AH V G R AW N D T R AE N S P ER T EY SH AH N IH Z AH V EY L AH B AH L IH N D EH N V ER	atis_ground_service
i want to hear a joel hastings melody	AY W AO N T T UW HH IY R EY JH OW AH L HH EY S T IH NG Z M EH L AH D IY	PlayMusic
show movie schedules for douglas theatre company	SH OW M UW V IY S K EH JH UW L Z F ER D AH G L AH S TH IY AH T ER K AH M P AH N IY	SearchScreeningEvent
go to the photograph the inflated tear	G OW T UW DH AH F OW T AH G R AE F DH AH IH N F L EY T IH D T EH R	SearchCreativeWork

Table IV shows the example of utterances from ATIS and Snips. The first column is the text representation of the utterance. The second column is phoneme representation of the utterance generated from g2p-seq2seq[16] model as discussed earlier. The third column 'label' is the correct label to be predicted for the utterance. In phoneme representation column we are showing phonemes space separated but in actually experiment space was considered after every word's phoneme representation.

For experiment, same architecture as shown in Figure 4 was developed using Keras. Model was run for 100 epochs, although models converge before 100th epoch. Batch size for experiment was taken as 64. We used 'adam' optimizers and 'categorical cross entropy' as loss function for the models.

IV. EVALUATION AND DISCUSSIONS

Table V, VI and VII shows top 1, top 2 and top 3 classification accuracy on two datasets with Model 1 (where weights are not shared) and Model 2 (where weights are being shared). Although in end-to-end we propose top k (where k=3) labels to use for LM Adaptation but to compare with state-of-the-art text based models, we are using top-1 results. From the Table V, VI and VII it is clear that on increasing the phoneme length the accuracy increases. Even after giving few phonemes (15 phonemes) at stage one model (Considering top 1 label and Model 1) is able to achieve 85.11% accuracy in ATIS and 91% in case of Snips whereas at third stage model is able to predict 95.18% in case of ATIS and 96.29% in case of Snips. The results of all the three tables V, VI and VII are convincing for the fact that with partial phoneme input model is able to converge and it worked similar to text based models.

TABLE V. ACCURACY ON TWO DATASETS CONSIDERING TOP-1 LABELS USING MODEL 1 (NON SHAREABLE WEIGHTS) AND MODEL 2 (SHAREABLE WEIGHTS)

Top 1 Labels				
Phoneme Input	***Model 1***		***Model 2***	
	ATIS	***Snips***	***ATIS***	***Snips***
15 Phoneme Input	85.11	91	84.77	89.86
25 Phoneme Input	92.05	94.71	89.59	93.86
40 Phoneme Input	95.18	96.29	90.82	96.29

TABLE VI. ACCURACY ON TWO DATASETS CONSIDERING TOP-2 LABELS USING MODEL 1 (NON SHAREABLE WEIGHTS) AND MODEL 2 (SHAREABLE WEIGHTS)

Model 1 – Top 2 Labels				
Phoneme Input	***Model 1***		***Model 2***	
	ATIS	***Snips***	***ATIS***	***Snips***
15 Phoneme Input	90.26	96.86	90.82	96.12
25 Phoneme Input	95.74	98.43	94.29	98.29
40 Phoneme Input	97.2	99.29	95.63	98.71

TABLE VII. ACCURACY ON TWO DATASETS CONSIDERING TOP-3 LABELS USING MODEL 1 (NON SHAREABLE WEIGHTS) AND MODEL 2 (SHAREABLE WEIGHTS)

Model 1 – Top 3 Labels				
	Model 1		*Model 2*	
	ATIS	*Snips*	*ATIS*	*Snips*
15 Phoneme Input	92.05	98.29	91.71	98.14
25 Phoneme Input	96.42	99.71	95.3	99.57
40 Phoneme Input	**97.76**	**99.57**	95.63	99.29

A. Model 1 vs Model 2

From Table V, VI and VII we can compare the accuracies between the two types of proposed models. Although Model 1 where weights are not being shared performed better but the difference between Model 1 and Model 2 is very small. In case of snips dataset the accuracy with Model 1 and Model 2 is almost similar (Table VII). In both the models on increasing the phoneme length the accuracy is improved. In terms of memory and parameter optimisation, Model 2 is better than Model 1.

B. MultiStage vs State-of-the-art Techniques

In this section we compared our best model accuracy with recent state-of-the-art models. The state-of-the-art models are based on text input whereas our model is based on partial phoneme input and not on full sentence. The results considering top 1 are close to state-of-the-art as shown in Table VIII. If we consider top 3 we surpasses state-of-the-art in case of ATIS as well as Snips.

TABLE VIII. STATE-OF-THE-ART COMPARISION

State-of-the-art comparison		
Models	*ATIS*	*Snips*
15 Phoneme Input (Model 1 - Top 1)	85.11	91
25 Phoneme Input (Model 1 - Top 1)	92.05	94.71
40 Phoneme Input (Model 1 - Top 1)	**95.18**	**96.29**
15 Phoneme Input (Model 1 - Top 3)	92.05	98.29
25 Phoneme Input (Model 1 - Top 3)	96.42	99.71
40 Phoneme Input (Model 1 - Top 3)	**97.76**	**99.57**
Attention-based RNN [21], 2016	91.1	97.0
Bi-Directional RNN-LSTM [22], 2016	92.6	96.9
Slot-Gated (Full Attention) [12], 2018	93.6	97.0
Slot-Gated (Intent Attention) [12] , 2018	94.1	96.8
Attention-Based CNN-BLSTM [22], 2018	97.17	-
Parallel Intent and Slot (Model 1) [23], 2019	96.87	98.14
Parallel Intent and Slot (Model 2) [23], 2019	97.42	98.14

C. Discussions

Domain prediction from partial phoneme output can be used for choosing top K Language models for rescoring word lattice. It helps in better selection among homophones belonging to different domains. As domain prediction can be done on partial phonemes, it does not add more time delay in ASR processing. Accuracy of phoneme-based models is very close to text based state-of-the-art models. We also observed that correlation between output labels of phoneme-based model is better than text based model and it is very important for rescoring word lattice. In general, feature space of phoneme representation is richer than text, so phoneme based classification models work good on some of NLP problems like sentiment analysis [17]. We can also pass predicted domain from last ASR frame to NLU thereby reducing the sub-task of NLU of domain prediction.

D. Future scope

Proposed network works well with partial phonemes, but at train time phonemes are generated using g2p-seq2seq [16] model and test time AM will generate phoneme sequence, so there will be some differences in Accuracy. This is one of reason we are using top K domain prediction for rescoring lattice. However, as future task we wish to benchmark partial phonemes based models, partial text based models or both for domain prediction. Further, we will work on more sophisticated ways of using domain specific LMs and present ASR results. Phoneme can also be used for intent and slot prediction that can help in building unified ASR-NLU system [24-25] as shown in Figure 6, bypassing conversion of voice to text, reducing the latency in voice systems.

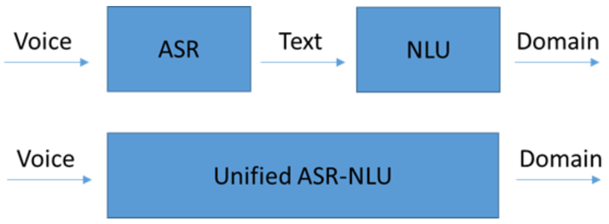

Fig. 6. Unified ASR-NLU System.

V. CONCLUSIONS

With proposed multistage CNN architecture system, which can process partially generated phoneme sequence for domain prediction. We are able to achieve accuracy similar to state-of-the-art text classification systems. There is significant improvement in accuracy between multiple stages, which process different length of phoneme sequence. It signifies that along with processing raw audio, classification accuracy is increasing and hence domain specific corrections will improve. Also for some domains like news and music there are frequent change in probability distribution of words, which can be incorporated in ASR by just updating domain specific Language Model.

VI. ACKNOWLEDGMENT

The authors would like to acknowledge the support of Samsung R&D Institute-India, Bangalore in this work.

REFERENCES

[1] P. Daniel, G. Arnab, B. Gilles, B. Lukas, G. Ondrej, G. Nagendra, H. Mirko, M. Petr, Q. Yanmin, S. Petr, S. Jan, S. Georg, V. Karel The Kaldi speech recognition toolkit. IEEE Signal Processing Society, 2011.

[2] Y.N. Chen, D. H. Tur, G. Tur, J. Gao and L. Deng, "End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding," Interspeech, 2016.

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

[4] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho and Y. Bengio "Attention-based models for speech recognition." Advances in neural information processing systems. 2015.

[5] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016.

[6] L. F. D'Haro and R. E. Banchs. "Automatic correction of ASR outputs by using machine translation," Interspeech 2016 : 3469-3473.

[7] Y. Bassil, M. Alwani. "Post-editing error correction algorithm for speech recognition using bing spelling suggestion." arXiv preprint arXiv:1203.5255 (2012).

[8] R. Errattahi, S. Deena, A. E. Hannani and H. Ouahmane "Improving ASR Error Detection with RNNLM Adaptation," IEEE Spoken Language Technology Workshop (SLT). IEEE, 2018.

[9] R. Errattahi, A. EL. Hannani, T. Hain and H. Ouahmane "System-independent ASR error detection and classification using Recurrent Neural Network," Computer Speech & Language 55 (2019): 187-199.

[10] R. Errattahi, A. EL. Hannani, T. Hain and H. Ouahmane. "Towards a generic approach for automatic speech recognition error detection and classification;" 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP). IEEE, 2018.

[11] Shivakumar, P. Gurunath, H. Li, K. Knight, and P. Georgiou "Learning from past mistakes: improving automatic speech recognition output via noisy-clean phrase context modeling," APSIPA Transactions on Signal and Information Processing 8 (2019).

[12] Guo, Jinxi, T. N. Sainath, and R. J. Weiss. "A spelling correction model for end-to-end speech recognition." arXiv preprint arXiv:1902.07178 (2019).

[13] Sagae, Kenji, G. Christian, D. DeVault, and D. Traum. "Towards natural language understanding of partial speech recognition results in dialogue systems." Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers. 2009.

[14] Traum, David, D. DeVault, J. Lee, Z. Wang, and S. Marsella. "Incremental dialogue understanding and feedback for multiparty, multimodal conversation," International Conference on Intelligent Virtual Agents. Springer, Berlin, Heidelberg, 2012.

[15] T. Mikolov, K. Chen, G. Corrado, & J. Dean, "Efficient Estimation of Word Representations in Vector Space," In Proceedings of Workshop at ICLR, 2013

[16] https://github.com/cmusphinx/g2p-seq2seq.

[17] P. Yenigalla, A. Kumar, S. Tripathi, C. Singh, S. Kar and J. Vepa. "Speech Emotion Recognition Using Spectrogram & Phoneme Embedding," Interspeech 2018: 3688-3692. .

[18] Y. Kim. "Convolutional neural networks for sentence classification," Proceedings of the 2014 Conference on EMNLP, pp. 1746–1751, 2014

[19] Goo, C. Wen, G. Gao, Y. K. Hsu, C. L. Huo, T. C. Chen, K. W. Hsu, and Y. N. Chen. "Slot-gated modeling for joint slot filling and intent prediction," Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). Vol. 2. 2018.

[20] Wang, Yufan, L. Tang, and T. He. "Attention-Based CNN-BLSTM Networks for Joint Intent Detection and Slot Filling," Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer, Cham, 2018. 250-261.

[21] Liu, B., Ian, L.: Attention-based recurrent neural network models for joint intent detection and slot filling. arXiv :1609.01454 (2016).

[22] D. Hakkani, G. Tur, A. Celikyilmaz, Y.N. Chen, L. Deng, Y. Wang.: Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM. In Interspeech 2016.

[23] A. Bhasin, B. Natarajan, G. Mathur, J. H. Jeon, and J. Kim. "Unified Parallel Intent and Slot Prediction with Cross Fusion and Slot Masking." In International Conference on Applications of Natural Language to Information Systems, pp. 277-285. Springer, Cham, 2019.

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

[25] L Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio. (2019). Speech model pre-training for end-to-end spoken language understanding. arXiv preprint arXiv:1904.03670.