Transfer Learning Based Free-Form Speech Command Classification for Low-Resource Languages

Yohan Karunanayake
University of Moratuwa
Sri Lanka
yohan.13@cse.mrt.ac.lk

Uthayasanker Thayasivam
University of Moratuwa
Sri Lanka
rtuthaya@uom.lk

Surangika Ranathunga
University of Moratuwa
Sri Lanka
surangikar@uom.lk

Abstract

Current state-of-the-art speech-based user interfaces use data intense methodologies to recognize free-form speech commands. However, this is not viable for low-resource languages, which lack speech data. This restricts the usability of such interfaces to a limited number of languages. In this paper, we propose a methodology to develop a robust domain-specific speech command classification system for low-resource languages using speech data of a high-resource language. In this transfer learning-based approach, we used a Convolution Neural Network (CNN) to identify a fixed set of intents using an ASR-based character probability map. We were able to achieve significant results for Sinhala and Tamil datasets using an English based ASR, which attests the robustness of the proposed approach.

.

1 Introduction

Speech command recognizable user interfaces are becoming popular since they are more natural for end-users to interact with. Google Assistant[1], and Amazon Alexa[2] can be highlighted as few such commercial services, which are ranging from smartphones to home automation. These are capable of identifying the intent of free-form speech commands given by the user. To enable this kind of service, Automatic Speech Recognition (ASR) systems and Natural Language Understanding (NLU) systems work together with a very high level of accuracy (Ram et al., 2018).

If ASR or NLU components have suboptimal results, it directly affects the final output (Yaman et al., 2008; Rao et al., 2018). Hence, to get good results in ASR systems, it is common to use very large speech corpora (Hannun et al., 2014; Amodei et al., 2016; Chiu et al., 2018). However, low-resource languages (LRL) do not have this luxury. Here, languages that have a limited presence on the Internet and those that lack electronic resources for speech and/or language processing are referred to as low-resource languages (LRLs) (Besacier et al., 2014). Because of this reason despite the applicability, speech-based user interfaces are limited to common languages. For LRLs researchers have focused on narrower scopes such as recognition of digits or keywords (Manamperi et al., 2018; Chen et al., 2015). However, free-form commands are difficult to manage in this way since there can be overlappings between commands.

Buddhika et al. (2018); Chen et al. (2018) show some direct speech classification approaches to its intents. In particular, Buddhika et al. (2018) have given some attention for the low resource setting. Additionally, Transfer learning is used to exploit the issue of limited data in some of the ASR based research (Huang et al., 2013; Kunze et al., 2017).

In this paper, we present an improved and effective methodology to classify domain-specific free-form speech commands while utilizing this direct classification and transfer learning approaches. Here, we use a character probability map from an ASR model trained on English to identify intents. Performance of this methodology is evaluated using Sinhala (Buddhika et al., 2018) and newly collected Tamil datasets. The proposed approach can reach to a reasonable accuracy using limited training data.

Rest of the paper is organized as follows. Section 2 presents related work, section 3 describes methodology used. Section 4 and 5 provides details of the datasets and experiments. Section 6 presents a detailed analysis of the obtained results. Finally Section 7 concludes the paper.

[1] https://assistant.google.com
[2] https://developer.amazon.com/alexa

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 288–294
Florence, Italy, July 28 - August 2, 2019.

2 Related Work

Most of the previous research has used separate ASR and NLU components to classify speech intents. In this approach, transcripts generated from the ASR module are fed as input for a separate text classifier (Yaman et al., 2008; Rao et al., 2018). Here, an erroneous transcript from the ASR module can affect the final results of this cascaded system (Yaman et al., 2008; Rao et al., 2018). In this approach, two separately trained subsystems are connected to work jointly. As a solution for these issues, Yaman et al. (2008) proposed a joint optimization technique and use of the n-best list of the ASR output. Later He and Deng (2013) extended this work by developing a generalized framework. However, these systems require a large amount of speech data, corresponding transcript, and their class labels. Further, the ASR component used in these systems requires language models and phoneme dictionaries to function, which are difficult to find for low-resource languages.

This cascading approach is effective when there is a highly accurate ASR in the target language. Rao et al. (2018) present such a system to navigate in an entertainment platform for English. Here, they have used a separate ASR system to convert speech into text. More importantly, they highlight that a lower performance of ASR affects the entire system.

More recently, researchers have presented some approaches that aim to go beyond cascading ASR components. In this way, they have tried to eliminate the use of intermediate text representations and have used automatically generated acoustic level features for classification. Liu et al. (2017) proposed topic identification in speech without the need for manual transcriptions and phoneme dictionaries. Here, the input features are bottleneck features extracted from a conventional ASR system trained with transcribed multilingual data. Then these features are classified through CNN and SVM classifiers. Additionally Lee et al. (2015) have highlighted that effectiveness of this kind of bottleneck features of speech when comparing different speech queries.

Chen et al. (2018); Buddhika et al. (2018) present two different direct classification approaches to determine the intent of a given spoken utterance. Chen et al. (2018) have used a neural network based acoustic model and a CNN based classifier. However, this requires transcripts of the speech data to train the acoustic model, thus accuracy depends on the availability of a large amount of speech data. One advantage of this approach is that we can optimize the final model once we combined the two models. Buddhika et al. (2018) classified speech directly using MFCC (Mel-frequency Cepstral Coefficients) of the speech signals as features. In this approach, they have used only 10 hours of speech data to achieve reasonable accuracy.

3 Methodology

In section 2, we showed that research work of Liu et al. (2017); Chen et al. (2018); Buddhika et al. (2018) has benefited from direct speech classification approach. Additionally, as shown in the work of Lee et al. (2015); Liu et al. (2017), it is beneficial to use automatically discovered acoustic related features. Therefore our key idea is reusing a well trained ASR neural network on high resource language as a feature transformation module. This is known as transfer learning (Pan and Yang, 2010). Here, we try to reuse the knowledge learned from one task to another associated task. Current well trained neural network based end-to-end ASR models are capable of converting given spoken utterance into the corresponding character sequence. Therefore these ASR models can convert speech into some character representation. Our approach is to reuse this ability in low-resource speech classification.

We used DeepSpeech (DS) (Hannun et al., 2014) model as the ASR model. DS model consists of 5 hidden layers including a bidirectional recurrent layer. Input for the model is a time-series of audio features for every timeslice. MFCC coefficients are used as features. Model converts this input sequence $x^{(i)}$ into a sequence of character probabilities $y^{(i)}$, with $\hat{y}_t = \mathbb{P}(c_t|x)$, where $c_t =\in \{a, b, c, .., z, space, apostrophe, blank\}$ in English model. These probability values are calculated by a softmax layer. Finally, the corresponding transcript is generated using the probabilities via beam search decoding with or without combining a language model.

Here, we selected intermediate probability values as the transfer learning features from the model. Any feature generated after this layer is ineffective since it is affected by the beam search and it only outputs the best possible character sequence. Before the final softmax layer, there is a

bi-directional recurrent layer, which is very critical for detecting sequence features in speech. Without this layer, the model is useless (Hannun et al., 2014; Amodei et al., 2016). Hence, the only possible way to extract features is after the softmax layer. Additionally, this layer provides normalized probability values for each time step. Figure 1 shows a visualization of this intermediate character probability map for a Sinhala speech query containing 'ශේෂය කියද - śēṣaya kīyada'.

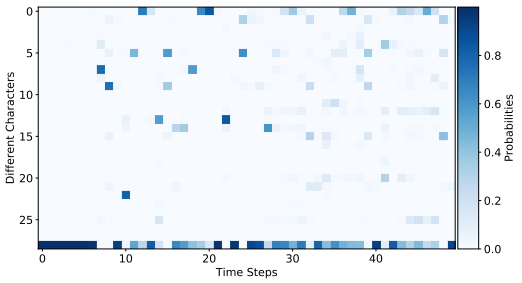

Figure 1: Visualization of probability output for Sinhala utterance

In this considering scenario, we need to identify a fixed set of intents related to a specific domain. Instead of converting these probability values into a text representation, we classify these obtained features directly in to intents as in (Liu et al., 2017; Chen et al., 2018). We experimented with different classifier models such as Support Vector Machines (SVM), Feed Forward Networks (FFN), which used in previous works. Further, in the work of Liu et al. (2017); Chen et al. (2018), they have shown the effectiveness of Convolutional Neural Networks - CNN to classify intermediate features of the speech. Because of this, we evaluated the performance of CNN. Additionally, We examined the effectiveness of 1-dimensional(1D) and 2-dimensional(2D) convolution for feature classification. Figure 2 shows the architecture of the final CNN based model. Please refer to 'Supplementary Material' for the detail of model parameters.

4 Datasets

We used two different free-form speech command datasets to measure the accuracy of the proposed methodology. The first one is a Sinhala dataset and contains audio clips in the banking domain (Buddhika et al., 2018). Since it was difficult to find such other datasets for low-resource languages, we created another dataset in the Tamil language,

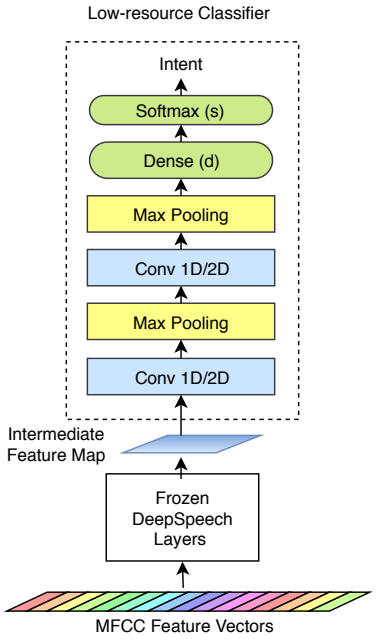

Figure 2: Architecture of the final model

which contains the same intentions as Sinhala dataset. Both Sinhala and Tamil are morphologically different languages. Table 1 summarizes the details.

Intent	Sinhala		Tamil	
	I	S	I	S
1. Request Acc. balance	8	1712	7	101
2. Money deposit	7	1306	7	75
3. Money withdraw	8	1548	5	62
4. Bill payments	5	1004	4	46
5. Money transfer	7	1271	4	49
6. Credit card payments	4	795	4	67
Total	39	7624	31	400
Unique words	32		46	

Table 1: Details of the data sets (I-Inflections, S-Number of samples)

Original Sinhala dataset contained 10 hours of speech data from 152 males and 63 females students in the age between 20 to 25 years. We had to revalidate the dataset since it included some miss-classified, too lengthy and erroneous speech queries. The final data set contained 7624 samples totaling 7.5 hours. Tamil dataset contains 0.5 hours of speech data from 40 males and females students in the same age group. There were 400 samples in the Tamil dataset. The length of each audio clip is less than 7 seconds.

5 Experiments

For the transfer learning task, we considered the DeepSpeech (DS) model 1 (Hannun et al., 2014).

	Benchmark		Current			
Approach	SVM	6L FFN	TL + SVM	TL + FFN	TL + 1D CNN	TL + 2D CNN
Features	MFCC		DS Intermediate			
Accuracy Sinhala	48.79%	63.23%	70.04%	74.67%	**93.16%**	92.09%
Accuracy Tamil	29.25%	26.98%	23.77%	35.50%	37.57%	**76.30%**

Table 2: Summary of results with different approaches and overall accuracy values

This model and some other neural network based ASR modes provide a probability map for each character in each time step. Due to high computational demand for training, we adopted an already available pre-trained DS model by Mozilla[3]. This model uses the first 26 MFCC features as input. Model is trained on American English and achieves an 11% word error rate on the LibriSpeech clean test corpus.

Given the DS English model, we extract the intermediate probability features for a given speech sample and then fed them into the classifier. Further, we employed a Bayesian optimization based algorithm for hyperparameter tuining (Bergstra et al., 2013). Since datasets are small we used 5 fold cross-validation to evaluate the accuracy.

We selected method presented in (Buddhika et al., 2018) as our benchmark. In their work, they have used the first 13 MFCC features as input for the SVM, FFN classifiers. Since we had to validate the Sinhala dataset, we reevaluate the accuracy values on the validated dataset using 5-fold cross-validation. Additionally, we performed the same experiments on newly collected Tamil dataset to examine the language independence of the proposing method. Table 2 summarizes the outcomes of these different approaches. In all experiments, class distribution among all data splits was nearly equal.

In this work, we are concerned about the amount of available data. Hence, we evaluated the accuracy change of the best performing approaches with the size of training samples. We perform this on the Sinhala dataset since it has more than 4000 data samples. We drew multiple random samples with a particular size and performed 5-fold cross-validation. Here, the number of random samples is 20. Table 3 summarizes the experiment results.

In another experiment, we examined the end-to-end text output of the DS English model for a given Sinhala speech query. Table 4 presents some of these outputs.

[3] https://github.com/mozilla/DeepSpeech

6 Result and Discussion

We were able to achieve 93.16% and 76.30% overall accuracy for Sinhala and Tamil datasets respectively using 5-fold cross-validation. Table 2 provides a comparison of previous and our approaches. It shows clearly that the proposed method is more viable than the previous direct speech feature classification approach. One possible reason can be the reduction of noise in speech signals. In this situation, the DS model is capable of removing these noises since it is already trained on noisy data. Another reason is that reduction of the feature space. Additionally, in this way, we can have more accurate results using small dataset.

Intent	Sinhala			Tamil		
	F1	P	R	F1	P	R
1	0.96	0.94	0.99	0.87	0.89	0.87
2	0.93	0.97	0.89	0.80	0.78	0.84
3	0.91	0.87	0.95	0.75	0.89	0.66
4	0.89	0.93	0.87	0.64	0.75	0.63
5	0.96	0.97	0.95	0.60	0.76	0.51
6	0.92	0.95	0.89	0.79	0.74	0.89
Average	0.93	0.93	0.93	0.76	0.81	0.76

Table 3: Classification results of best performing models (F1- F1-Score, P- Precision, R- Recall)

Table 3 shows the averaged precision, recall and F1-score values for each intent class and two datasets. In the Sinhala dataset, all classes achieve more than 0.9 F1-score, except for type 4 intent. Type 1 intent shows the highest F1-score among all and, this must be because of the higher number of data samples available for this class. Despite that, type 6 intent also reports 0.93 f1-score even with a lower number of data samples. Tamil data shows a slightly different result. Intent types 4,5 report the lowest score in the Tamil dataset and the number of speech queries from these classes are comparatively low in the dataset. Further, we can observe that the Tamil classifier is incapable of accurately identifying positive intent classes 4 and 5 (since lower recall value).

Compared to Sinhala data with a sample size of 500, Tamil dataset reports high overall accuracy with 400 samples. Tamil dataset contains

codemixed speech quires since it is more natural when in speaking. These words are in English. Additionally, the feature generator model (DS model) is also trained in English data. This can result in more overall accuracy in Tamil data set. Additionally, type 6 intent commands contain English words in both datasets and this can result for higher precision value.

Further, sentences with more overlapping words with other sentences (different intent type) and with limited length tend to misclassify more. Hence classes, type 3,4 in Sinhala, type 2,4 in Tamil dataset show lower accuracy.

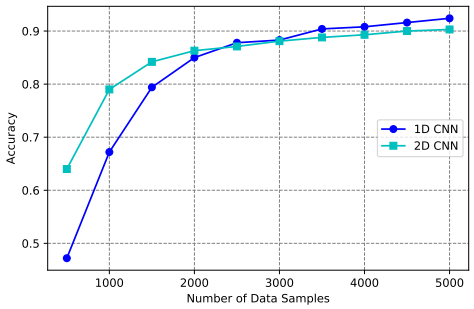

Figure 3: CNN classifier accuracy variance with the number of samples (Sinhala dataset)

Figure 3 summarize the overall accuracy change of best performing classifiers with samples size. As it shows having 1000 samples is enough to achieve nearly 80% overall accuracy. After that, it reaches saturation. Furthermore, it reports 77% overall accuracy for Tamil dataset with 320 training samples. This highlights the effectiveness of the proposed transfer learning approach in limited data situations.

Additionally, Figure 3 shows the most effective CNN model type with the number of available data samples to classify sequential feature maps. As it shows, it is useful to use 2D CNN based classifiers when there is a very limited amount of data. However, when there are relatively more data (More than 4000 samples in Sinhala dataset) 1D CNN based classifiers gives higher results. We can see this effect on Tamil dataset also. As table 2 shows 1D CNN model accuracy is low compared to 2D CNN model with 400 data samples.

Further, we examined the speech decoding capability of the English model. See Table 4. Here 'Utterance' is the pronounced Sinhal sentence, 'Eng. Transcript' is the ideal English transcript. 'DS output' lists the generated transcripts from the full model. In these generated outputs, the first few characters are decoded correctly. But, in the latter part, this decoding is compromised by the possible character sequences of the English language since it is trained in English. From this, we can infer that this character probability map is closer to text representation than the MFCC features. Hence, this can improve the classification accuracy.

Utterance	ශේෂය කීයද	ඉතිරිය කීයද
Eng. Transcript	'sheshaya keeyada'	'ithiriya keeyada'
DS Output	'**she s** reci ete' '**sheis h**eki edit' '**sheis** ae an'	'**it** cillety edet' '**it ti**a gaviade' '**it** lid en'

Table 4: DS transcript for some Sinhala utterances

7 Conclusion

In this study, we proposed a method to identify the intent of free-form commands in a low-resource language. We used an ASR model trained on the English language to classify the Sinhala and Tamil low-resource datasets. The proposed method outperforms previous work and, even with a limited number of samples, it can reach to a reasonable accuracy.

CNN base classifiers perform well in the classification of character probability maps generated by ASRs. Further, 1D CNN models work better with a higher number of samples, while 2D CNN models work better with a small amount of data. In the future, we plan to extend this study by incorporating more data from different languages and domains.

Acknowledgments

This research was funded by a Senate Research Committee (SRC) Grant of the University of Moratuwa. We thank for the support received from the LK Domain Registry. We thank Mr. P. Thananjay for assistance with data validation.

References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182.

James Bergstra, Dan Yamins, and David D Cox. 2013. Hyperopt: A python library for optimizing the hy-

perparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, pages 13–20. Citeseer.

Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100.

Darshana Buddhika, Ranula Liyadipita, Sudeepa Nadeeshan, Hasini Witharana, Sanath Javasena, and Uthayasanker Thayasivam. 2018. Domain specific intent classification of sinhala speech data. In *2018 International Conference on Asian Language Processing (IALP)*, pages 197–202. IEEE.

Nancy F Chen, Chongjia Ni, I-Fan Chen, Sunil Sivadas, Haihua Xu, Xiong Xiao, Tze Siong Lau, Su Jun Leow, Boon Pang Lim, Cheung-Chi Leung, et al. 2015. Low-resource keyword search strategies for tamil. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5366–5370. IEEE.

Yuan-Ping Chen, Ryan Price, and Srinivas Bangalore. 2018. Spoken language understanding without speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6189–6193. IEEE.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Xiaodong He and Li Deng. 2013. Speech-centric information processing: An optimization-oriented approach. *Proceedings of the IEEE*, 101(5):1116–1135.

Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7304–7308. IEEE.

Julius Kunze, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. 2017. Transfer learning for speech recognition on a budget. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 168–177.

Lin-shan Lee, James Glass, Hung-yi Lee, and Chun-an Chan. 2015. Spoken content retrievalbeyond cascading speech recognition with text retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(9):1389–1420.

Chunxi Liu, Jan Trmal, Matthew Wiesner, Craig Harman, and Sanjeev Khudanpur. 2017. Topic identification for speech without asr. In *Proc. Interspeech 2017*, pages 2501–2505.

Wageesha Manamperi, Dinesha Karunathilake, Thilini Madhushani, Nimasha Galagedara, and Dileeka Dias. 2018. Sinhala speech recognition for interactive voice response systems accessed through mobile phones. In *2018 Moratuwa Engineering Research Conference (MERCon)*, pages 241–246. IEEE.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.

Jinfeng Rao, Ferhan Ture, and Jimmy Lin. 2018. Multi-task learning with neural networks for voice query understanding on an entertainment platform. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 636–645. ACM.

Sibel Yaman, Li Deng, Dong Yu, Ye-Yi Wang, and Alex Acero. 2008. An integrative and discriminative technique for spoken utterance classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1207–1214.

A Supplemental Material

Table 5 present hyperparameters for low-resourced models described in the section 3

	Sinhala Models		Tamil Models	
Layer	1D CNN	2D CNN	1D CNN	2D CNN
1. Conv	Filters 38 Kernel Size 19	Filters 16 Kernel Size 1x8	Filters 39 Kernel Size 18	Filters 14 Kernel Size 5x1
2. Max Pooling	Size 18 Stride 7	Size 6x1 Stride 5x5	Size 25 Stride 5	Size 13x1 Stride 5x1
3. Conv	Filters 28 Kernel Size 22	Filters 17 Kernel Size 20x8	Filters 26 Kernel Size 19	Filters 13 Kernel Size 11x20
4. Max Pooling	Size 22 Stride 10	Size 19x2 Stride 16x8	Size 20 Stride 5	Size 17x1 Stride 2x7
5. Dense	Units 131	Units 118	Units 84	Units 127
6. Softmax	6	6	6	6

Table 5: Hyperparameters for CNN classifier models