RnD Endterm Report

D.Chandra Sekhar
190050031

May 11, 2022

Topic: Targeted Subset Selection for Accented Speech Recognition

Guide: Ganesh RamaKrishnan Sir

Co-Guide: Preethi Jyothi Madam

Contents

1 Problem Statement

1.1 Idea

Recent ASR models have achieved good WERs on **native** English speech. But still they fail to achieve good WERs on **accented** English speech. The reason for this is : lack of good representation of accented speech samples in the training data-sets.

One solution for this problem is to just collect more speech samples of the targeted accent and finetune the pretrained model on those samples. But the limitation is that: collecting large number of such annotated/transcribed speech samples is expensive. Hence we seek for solutions which use comparitively lesser number of annotated speech samples.

1.2 Formal Statement

Name	Symbol	Meaning
Target Accent	$\mathbf{T}$	Accent on which we want the ASR model to perform well
Target Set	$\mathcal{T}$	Unannotated samples from target accent.
Ground Set	$\mathcal{G}$	Collection of large number of unannotated speech samples from various accents.
Selected Set	$\mathcal{X}$	$\mathcal{X} \subseteq \mathcal{G}$. Samples which are sent for annotation and further used for finetuning.
Test Set	$\mathcal{Y}$	Annotated Samples on which our finetuned model is evaluated.
Budget	$\mathcal{B}$	The maximum duration of speech samples which we can get annotated
Budget-Constraint	$\mathcal{C}$	$\sum_{x \in \mathcal{X}} \texttt{duration(x)} \leq \mathcal{B}$

Under the above notation, our goal is to propose an algorithm that selects $\mathcal{X}$ such that, $\mathcal{C}$ is obeyed and the performance of finetuned model on $\mathcal{Y}$ is maximised.

To evaluate the performance of our selection($\mathcal{X}$) we define an **Oracle** performance measure. It is calculated by sampling $\mathcal{Z}$ from $\mathcal{G}$ randomly s.t. $\mathcal{Z}$ only has samples from $\mathbf{T}$, annotating them, fine tuning our model on $\mathcal{Z}$ and evaluating finetuned model on $\mathcal{Y}$.

Under the experimental setting we actually know the accents of all samples in $\mathcal{G}$. Therefore sampling $\mathcal{Z}$ is possible. While sampling $\mathcal{X}$ we make sure that we don't use the accent information in any unrealistic way. In the real-world, we may not know accent information for all samples, but this is not a problem because, we anyway don't need to sample $\mathcal{Z}$ as it is just for checking the performance of our selections.

Although this formulation was done from the perspective of Accent-Personalisation, same notation and algorithms can be applied for Speaker Personalisation setting also.

2 Algorithm

2.1 Idea

Our goal is to select $\mathcal{X}$ such that samples in $\mathcal{X}$ resemble $\mathbf{T}$ in terms of accent. For this we need an embedding of speech utterances that preserves the accent information. Once we get these embeddings we need to choose samples $\mathcal{X}$ from $\mathcal{G}$ such that, they are as **diverse** as possible and as **relevant** as possible to the target set($\mathcal{T}$). To ensure this diversity and relevancy(a.k.a. query coverage) we use Sub-Modular-Information(S.M.I) functions.

So the main challenge left in this method is in:

1. Choosing the right embeddings that preserve accent information.
   (a) MFCC features
   (b) Accent-Classifer features.
2. Choosing the right S.M.I functions that balance the diversity and coverage.
   (a) FL2MI
   (b) GCMI
   (c) LogDMI

2.2 Prev Work

Anmol and Mayank have already explored the idea of using MFCC features along with FL2MI, GCMI, LogDMI in [1]. It was observed that these work really well for Speaker and Accent Personalisation tasks on data sets such as Indic and L2-Arctic. Using MFCC features, T-SNE plots obtained showed that speech samples from the same user clustered together very well.

2.3 Issues

- When MFCC features were applied on Mozilla-Common Voice data set, the T-SNE plots didn't show well formed clusters. Each cluster had speech samples from many different accents. This clearly showed that embeddings should be improved from the basic MFCC features in such a way that they capture accent-information.
- The reason MFCC features worked well with L2-Arctic and Indic data sets is that both these data sets have very small number of speakers per accent and the speech samples didn't have noise, loudness variations etc. Basically MFCC was capturing speaker features well when there is no noise in the data set and since number of speakers per accent is = 1 (in Indic data set) it appeared as if MFCC was doing well on Accent-Identification.

This is where my RnD work exactly started. We needed to come up with new embeddings which capture accent information well.

3 Work done in RnD

Our main aim is to find embeddings that capture accent-information well in spite of noise, loudness variations etc.

3.1 Semi-Supervised Accent Classifier

3.1.1 Idea

First we thought of training an accent-classifier in a Semi-Supervised fashion. We do few iterations where in each iteration we select few utterances based on their current pseudo labels and send them for accent-labelling and transcription. Then we add these to our base training set and retrain our model(ASR + accent classifier) using a joint learning approach. At test time, given a test utterance we can take the last layer values of the accent classifier as the accent embedding of that utterance. These accent embeddings clearly preserve the accent information. But due to **difficulty of accent-labelling a speech sample at a later point in time**, this method is difficult to apply in practice.

3.1.2 Papers read

- **Neural SSL for Text Classification**[2] : This paper compares various semi supervised learning approaches for the task of text classification.

  Let $\mathcal{U}$ denote in-domain **unlabelled** data, $\mathcal{D}$ denote in-domain **labelled** data. First a language model like `RoBERTa`(which can be initialised from scratch or from an open-domain trained version) is trained on $\mathcal{U}$, which is further fine-tuned on $\mathcal{D}$.

  Student-Teacher model is used to train text classifier. We use the embeddings from language model as inputs to the classifier. Teacher is trained on $\mathcal{D}$ and then is used to assign **pseudo-labels** to $\mathcal{U}$. Top-K samples from $\mathcal{U}$ w.r.t the confidence of teacher model on them are selected and we denote them as $\mathcal{D}'$. Now we have several alternatives in training the student model. Let $\mathcal{T}(\mathcal{X})$ denote pre-training of student model on data set $\mathcal{X}$ and let $\mathcal{F}(\mathcal{X})$ denote fine tuning of it on data set $\mathcal{X}$.

  Alternatives:

  1. $\mathcal{T}(\mathcal{D}')$
  2. $\mathcal{T}(\mathcal{D}+\mathcal{D}')\mathcal{F}(\mathcal{D})$
  3. $\mathcal{T}(\mathcal{D}')\mathcal{F}(\mathcal{D})$

  This process of training teacher and then using it to further train a student can be iterated. In the iterative method, student is used initialise the teacher for the next iteration then the teacher is finetuned on $\mathcal{D}$ and the process repeats.

  Now among these various alternatives (training language model and training student model) this paper evaluates all of those and the conclusions are as follows.

  Conclusion:

  1. Open domain + in-domain pre-training is best for language model when the model size is very large or the in-domain data is less.

2. Under the case of very large ($|\mathcal{U}|$) or smaller language model size, only in-domain pre-training is best for the language model.
3. Among various pseudo-labelling approaches for training student , $\mathcal{T}(\mathcal{D}+\mathcal{D}')$ works best when $|\mathcal{D}|$ is large and $\mathcal{T}(\mathcal{D}+\mathcal{D}')\mathcal{F}(\mathcal{D})$ works best otherwise.
4. When the best pretraining technique is combined with best pseudo-labelling approach, their performance gains add up and lead to an even better model.

3.2 Supervised Accent Classifier

3.2.1 Idea

Instead of training the accent classifier in a semi-supervised fashion we train it in a fully-supervised fashion. Our Accent Classifier uses pretrained Wav2Vec2.0 as its base model and adds an accent classifier layer on top of it. All layers except the last two layers of Wav2Vec2.0 are frozen. We used Mozilla-Common-Voice(MCV) dataset to train the model. Around 1000 samples each, from 8 different accents are used to train the classifier.

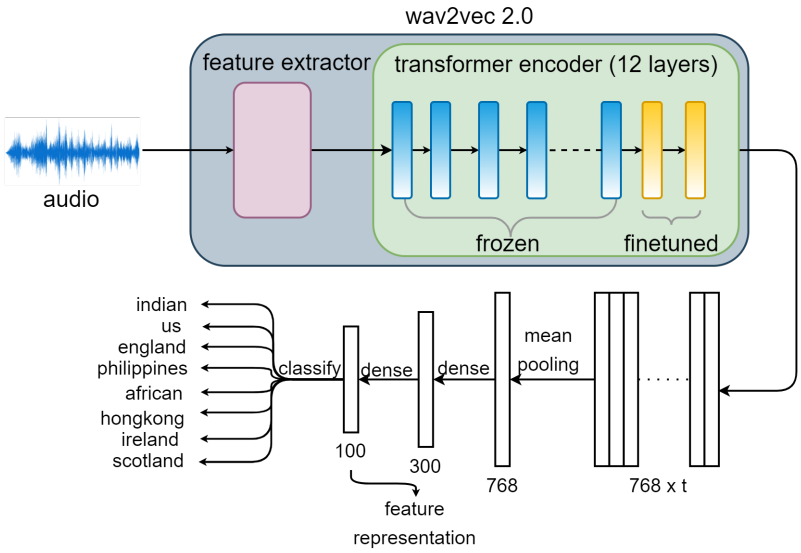


Figure 1: Classifier Architecture [1]

3.2.2 Papers Read

We wanted our classifier to generalise across unseen accents well. i.e. the classifier needs to predict a close-by accent when a sample which does not belong to any of the output accents is given as input. For this we explored Self-Supervised learning, Contrastive loss function based approaches. [3], [4], [5]

3.3 Entropy Based Selections

Though S.M.I. functions ensure coverage and diversity in the selected set $\mathcal{S}$, they don't **explicitly** try to select the samples on which our ASR model is most uncertain. But selecting these will help the ASR model to learn quicker using lesser number of samples.

[1] This image is drawn by me for this paper [1]

So to enforce this, we decided to do second-level selections on top of first level S.M.I. based selections. For this we explored the idea of entropy-based selections.

3.3.1 Papers-Read

- **Data selection for speech recognition**[6]: Paper's goal is to select utterances to train an acoustic model, in a sample-efficient manner. It assumes that all the utterances are already transcribed(i.e we work with true-transcripts). The proposed method is to select the utterances such that the cumulative distribution over words/phonemes/characters in the selected set is as close as possible to uniform distribution. This objective is rewritten in terms of KL-Divergence and is optimised greedily with no performance guarantees. Author reports that this method achieves WERs close to the model trained on full data.

- **Document Summarization technique for speech subset selection**: [7]
  This paper formulates subset selection as a monotone sub-modular optimisation problem. Monotone sub-modular functions can be optimised (upto a constant factor from the optimal solution) very efficiently using greedy strategies and moreover this is the best that can be done in **P**-time.

  This paper investigates `Facility Location function` and `Saturated Coverage function` together with the following similarity kernels:

  - **Fisher Similarity**: This is a purely acoustic similarity measure. Let $\theta_1, \dots \theta_m$ be the parameters of m-HMM models, then

$$\begin{aligned} U_X^{\theta} &= \nabla_{\theta} \texttt{log}(P(X|\theta) \\ U_X^{'} &= [(U_X^{\theta_1})^T, (U_X^{\theta_2})^T, \dots (U_X^{\theta_m})^T]^T \\ \texttt{d}_{i,j} &= ||U_i^{'} - U_j^{'}|| \\ \texttt{sim}_{i,j} &= (\max_{i',j'} d_{i',j'}) - d_{i,j} \end{aligned}$$

  - **TF-IDF weighted cosine similarity**: This is a discrete representation based similarity measure. We first construct pseudo-transcripts of each sentence. Now considering each transcript as a sentence with word being the 3-gram phonemes, we construct their TF-IDF vectors.

$$sim_{i,j} = \frac{< vec_i, vec_j >}{\sqrt{< vec_i, vec_i >< vec_j, vec_j >}}$$

  - **Gapped string kernel**: This is a discrete representations based similarity measure. Let Σ denote the set of phonemes, then $\phi : \Sigma^* \rightarrow \mathcal{H}$ is defined as

$$\phi_u^k(s) := \sum_{\mathbf{i}:u=s(i)} \lambda^{span(i)}, \quad u \in \Sigma^k$$

  The summation is over **i** which is the set of all sub-sequences over $\{1, \dots |s|\}$. Let $i = \{i_1, i_2, \dots i_q\}$ then $\texttt{span}(\mathbf{i}) = i_q - i_1 + 1$.

$$\mathcal{K}(s_i, s_j) = \frac{\sum_u < \phi_u^k(s_i), \phi_u^k(s_j) > w_u}{\sqrt{< vec_i, vec_i >< vec_j, vec_j >}} \quad u \in \Sigma^l$$

w_u is a weight dependent on the length of u. k, l, λ are hyper-parameters that are tuned on the validation set. More about the efficient computation and other details can be found here([8])

Conclusion: Gapped String kernel with facility location S.M.I function works the best and they produce much better results than random, entropy based baselines.

- **Unsupervised Submodular Subset Selection** [9]: The main contribution of this paper is in coming up with two novel sub modular functions. Then they train an unsupervised model that can produce pseudo-transcripts. Now they work on these pseudo transcripts, using the sub modular functions. For similarity based S.M.I functions, they use Gapped String kernel as similarity measure.

  Let **V** denote the ground set and **S** denote the selected subset, then

  Facility Location function:

$$f_{fac}(S) = \sum_{i \in \mathbf{V}} \max_{j \in \mathbf{S}} w_{i,j}$$

  $w_{i,j}$ can be any valid similarity metric . But this paper uses gapped string kernel as similarity measure 3.3.1. Facility location functions ensures coverage but may not ensure diversity enough. Hence they introduce **a novel diversity rewarding** monotonic sub-modular function.

  Diversity Reward function: Let $\mathbf{P}_1, \ldots \mathbf{P}_k$ denote disjoint partitions of the ground set. Then

$$f_{div}(S) = \sum_{n=1}^{K} \sqrt{\sum_{j \in P_n \cap S} \left(\sum_{i \in V} \frac{w_{i,j}}{|V|} \right)}$$

  This function encourages samples to be picked from all the partitions, hence encourages diversity.

  To get the benefits of both Facility Location and the Diversity based function, we consider the following S.M.I function

$$f_{fac+div}(S) = \lambda \times f_{div}(S) + (1 - \lambda) \times f_{fac}(S)$$

  λ is a hyper-parameter that is tuned on validation set.

  Issue: Both these S.M.I. functions are similarity based. So in order to find the optimal solution, we need to construct a graph, which would make the complexity at least quadratic. So we consider other class of S.M.I. functions, known as ***feature-based S.M.I functions***. **Feature Based S.M.I functions**

  - **Basic Idea**:

$$f_{fea}(S) = \sum_{u \in \mathcal{U}} g(m_u(S))$$

    Where $\mathcal{U}$ denotes the set of features, u denotes a particular feature, $m_u(S)$ denotes the ***amount*** of feature in set S, g is any concave function like `sqrt` .

– **Two-Layer S.M.I function**: Single Layer S.M.I functions enforce coverage of features as they are directly optimising that. But they don't capture redundancy between features. This paper introduces a **novel two-layer feature based S.M.I function** that can capture redundancy between features.

Let $\mathcal{U}^2$ denote the second level meta-features, $\mathcal{U}^1$ denote the first-level features. Let $|\mathcal{U}^2| = d_2$ and $|\mathcal{U}^1| = d_1$. Let $W \in \mathbf{R}^{d_2 \times d_1}$ be a matrix where $W(u_2, u_1)$ indicates the interaction between $u_2 \in \mathcal{U}^2$ and $u_1 \in \mathcal{U}^1$. Now we define the 2-layer S.M.I function as follows:

$$f_{\texttt{2-fea}} = \sum_{u_2 \in \mathcal{U}^2} g_1 \left(\sum_{u_1 \in \mathcal{U}^1} W(u_2, u_1) g_2(m_{u_1}(S)) \right)$$

- **N-best entropy**[10]: This paper deals with selecting a subset of the data in order to train an acoustic model. It assumes that a baseline model that can produce pseudo-transcripts for all utterances in ground set is already available. The proposed algorithm works on these pseudo-transcripts to do the subset selection. For any speech sample to be useful it needs to satisfy the following two criteria:

  1. **Informativeness**: Adding the sample should make the model learn as much as possible. So the utterances on which our model is least confident are the ones which are most *informative* for our model.

     Entropy Measure: Let u be the utterance whose entropy we need to measure. Let $< s_1, p_1 >, < s_2, p_2 >, \ldots < s_n, p_n >$ be the top-N transcriptions and their probabilities, ES_u be the entropy score of our acoustic model for utterance u.

     $$\texttt{ES}_\texttt{u} = -\sum_{i=1}^{n} \left(\left(\frac{p_i}{\sum_{j=1}^{n} p_j} \right) \texttt{log} \left(\frac{p_i}{\sum_{j=1}^{n} p_j} \right) \right)$$

     More the value of $\texttt{ES}_\texttt{u}$ more is its informativeness.

  2. **Representativeness**: Most of the times it happens that the samples on which our model is least confident are outliers i.e. ones which lie far away from the rest of the training data. According to informativeness these may be highly useful, but in reality they aren't. So to measure representativeness, we first construct n-gram based TF-IDF vectors for each pseudo-transcript and then calculate the distance of each TF-IDF vector from the mean TF-IDF vector where mean is taken over all the samples in the ground set. This distance is used to measure representativeness. Lesser the distance, more is its representativeness.

  Conclusion: Author reports that N-best entropy method beats other confidence based measures. When combined with the representativeness, the method improves even further.

- **Error Driven ASR Personalisation**[11]: This paper deals with the exact reverse of our problem, Given a ground set of lot of sentences, we try to pick a subset of sentences on which we need to collect the users speech utterance. This paper takes into account the following criteria:

1. **Informativeness**: To measure informativeness we try to calculate the error that our ASR model would make given an utterance corresponding to that sentence s. For this first we convert each grapheme(sentence) into its phonemes using a pre-trained grapheme-phoneme converter, then send these to a model that labels each phoneme with a probability value where the probability indicates the probability that the ASR model would predict that particular phoneme wrongly.
2. **Representativeness**: In order to encourage diversity among the selected set, we try to pick sentences which have phonemes that are **under-represented** in the sentences selected till then.

3.3.2 Coding-Work

Most of my coding work is focused on trying out entropy-based techniques on top of first level S.M.I selections, writing helping scripts for the team.

- **Listening Audio data**: When our team tried to calculate WER's of various techniques on the MCV(Mozilla Common Voice) dataset, they've found that pretrained model had very high WERs for Indian and African accents. Indian had WER ≈ 90 and African had WER ≈ 40. None of the papers that have worked on MCV have reported such results. So in-order to debug this issue, we thought of hearing to those samples manually and checking if the audio samples are really that unclear. Code for this can be found here.

  Conclusion: African samples are more like U.S samples with better English(in comparison to Indian samples). Also Indian samples had more noise than the African samples, but the noise was not that high. So the WER of around 90 was **not** justified. Later in a different experiment(about entropy) we observed that the predictions of our pretrained model on Indian samples are grossly incorrect and almost all Indian samples got the exact same wrong prediction. So this made it clear that our data-processing(converting .mp3 to .wav) had issues which led all utterances point to the same .wav file. The issue was in the script that converted all `.mp3` files to `.wav` files. Figuring out the pre-processing issue (accidentally :))and fixing it **is one of my major coding contribution to the project**.

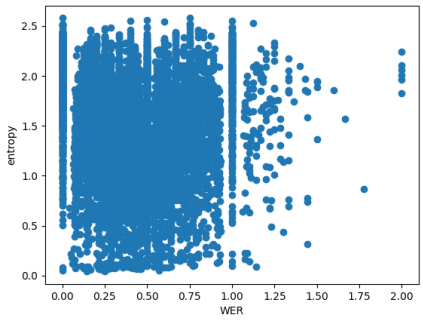
- **Frame Level Entropy Experiment**: We decided to implement the idea used in [10] on top of S.M.I selections. But unlike classical models, it is not easy to get top-N predictions from end-to-end deep models. So to get a measure of entropy, we used the following idea: Get the frame level entropy for each frame for the best prediction. Aggregate them using a suitable aggregating function to get the entropy for the utterance. Now we calculate the entropy values, for all the samples selected in the first level S.M.I selections. Then we pick the top-N of them and fine-tune our model on them. If this idea is working, then we should find that the W.E.R of model trained on top-N entropy based samples $<$ W.E.R of model trained on top-N samples based on S.M.I score. Code for this can be found here

  Unfortunately this didn't turn out to be the case. We tried with different aggregating functions to calculate the aggregated entropy from frame level entropies. We tried mean, median. But none of them worked.

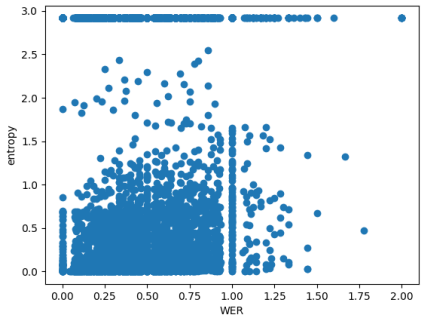
  WER values that we obtained for S.M.I + entropy, only S.M.I are compared here.

- **Correlation between W.E.R and Entropy**: As the above experiment didn't work, we tried to find what is wrong with our entropy criterion. For that we checked the correlation between W.E.Rs and the calculated entropies(both w.r.t our pretrained model). If our entropy measure is "really" an uncertainty measure, then we need to see a positive correlation W.E.R and entropy.

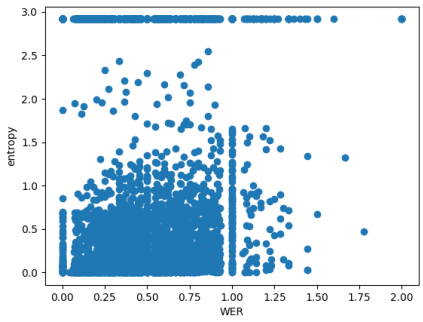
  Unfortunately this also did not turn out to be the case. We tried with various aggregating methods like mean, median, mean of top-100, median of top-100 to aggregate the frame level entropies. But in all cases we got almost 0 correlation between W.E.R and the entropy. Code for this can be found here.



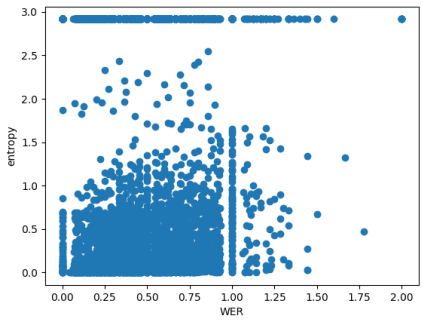
(a) Agg = Mean(Top 10 entropies)



(b) Agg = Median(Top 10 entropies)



(c) Agg = Median(Top 20 entropies)



(d) Agg = Median(Top 50 entropies)

  Clearly all figures show 0 correlation between aggregated entropy and W.E.R.

- **Checking if Content-based has any scope**: As the experiment with entropy based selection failed, we wanted to check whether content-based selection can have any scope. For this we wanted to compare the W.E.Rs of model finetuned with the following:

  1. Selecting **top-100** samples from S.M.I 800 selections.
  2. Selecting **random-100** samples from S.M.I 800 selections.

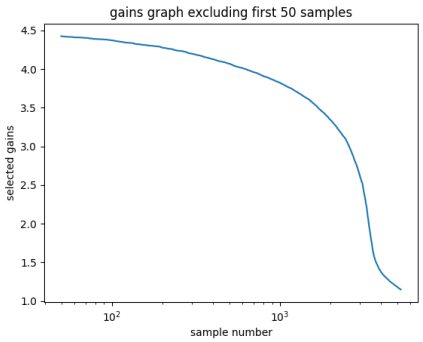
  If both of them have comparable W.E.Rs then we can expect the content-based selection to have some scope. If top-100 is significantly better than random-100 then we are good with S.M.I 100 itself(which is nothing but top-100 of S.M.I 800). Code for this can be found here and here. Results of this experiment can be found in the spreadsheet here.

  As the results suggest, both S.M.I - 100 and random-100 are comparable. This makes it clear that content-based selection can have some scope.

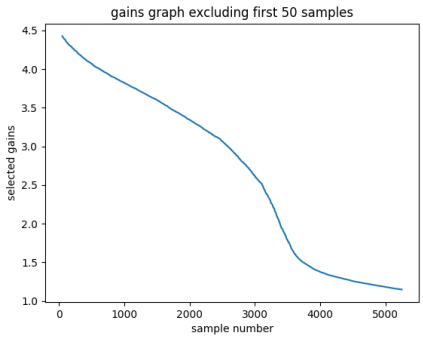
- **S.M.I scores for out of domain data**: Recall that we are doing content-based selection on top of S.M.I selections. So we want the S.M.I selections to select samples whose accent matches with that of target accent and content-based selections filter them and pick the samples that are going to make our model learn the most. As we are doing a two stage selection our budget of S.M.I should be higher now and also we need to take care that even with an increased budget we should not pick out of domain(accent) samples. So we need some signal to identify samples that are from different accent while doing S.M.I selections. For this we investigated if S.M.I scores would provide such a signal. We expected that S.M.I scores drop suddenly once all samples from the current accent are picked. To verify this we plotted graph of S.M.I score(y-axis) and the current serial number(x-axis). Code for this can be found here

  But the graphs did not show any such sudden drop when all the accents are used in the ground set, which means that S.M.I score can't be used as a signal. But when all the close by accents are removed from the ground-set, the graphs showed steep-decline.

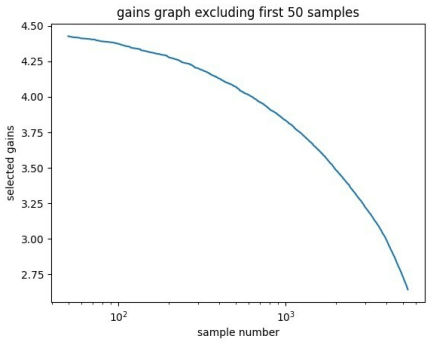
  Following are the graphs for `Accent: Malayalam Male` from Indic dataset.



(a) Close by accents removed. x-axis log scale



(b) Close by accents removed. x-axis usual scale



(c) All accents. x-axis log scale

3.4 Misc Papers Read

- **QuartzNet** [12] & [13]: We have explored this paper to understand the architecture of QuartzNet better. Primarily we studied about the semantics of output of the model, use of a Language model in the model.

  Conclusion: QuartzNet is an end-to-end model that does not use any language model. This also helps to compare various ASR models because their performance does not depend on what language model they use. But this obviously comes at the cost of our model making blunder grammatical mistakes when the utterance has noise. Regarding output: QuartzNet outputs a softmax over all the phonemes for each speech-frame.

- **Robust Submodular Observation Selection** [14]: This paper deals with Submodular observation selection under **multiple constraints**. It gives an algorithm that is at least as good as the optimal set but at a slightly increased budget. Further he proves that we can't do any better than this w.r.t. to the extra budget used. The algorithm is greedy in nature and can be implemented quite efficiently.

- Few other papers we referred include: [15], [16], [17]

References

[1] M. Kothyari, A. R. Mekala, R. Iyer, G. Ramakrishnan, and P. Jyothi, "Personalizing asr with limited data using targeted subset selection," *arXiv preprint arXiv:2110.04908*, 2021.

[2] Z. Sun, C. Fan, X. Sun, Y. Meng, F. Wu, and J. Li, "Neural semi-supervised learning for text classification under large-scale pretraining," *arXiv preprint arXiv:2011.08626*, 2020.

[3] T. Han, H. Huang, Z. Yang, and W. Han, "Supervised contrastive learning for accented speech recognition," *arXiv preprint arXiv:2107.00921*, 2021.

[4] K. Deng, S. Cao, and L. Ma, "Improving accent identification and accented speech recognition under a framework of self-supervised learning," *arXiv preprint arXiv:2109.07349*, 2021.

[5] A. Jain, M. Upreti, and P. Jyothi, "Improved accented speech recognition using accent embeddings and multi-task learning." in *Interspeech*, 2018, pp. 2454–2458.

[6] Y. Wu, R. Zhang, and A. Rudnicky, "Data selection for speech recognition," in *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*. IEEE, 2007, pp. 562–565.

[7] K. Wei, Y. Liu, K. Kirchhoff, and J. Bilmes, "Using document summarization techniques for speech data subset selection," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 721–726.

[8] J. Rousu, J. Shawe-Taylor, and T. Jaakkola, "Efficient computation of gapped substring kernels on large alphabets." *Journal of Machine Learning Research*, vol. 6, no. 9, 2005.

[9] K. Wei, Y. Liu, K. Kirchhoff, and J. Bilmes, "Unsupervised submodular subset selection for speech data," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4107–4111.

[10] N. Itoh, T. N. Sainath, D. N. Jiang, J. Zhou, and B. Ramabhadran, "N-best entropy based data selection for acoustic modeling," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4133–4136.

[11] A. Awasthi, A. Kansal, S. Sarawagi, and P. Jyothi, "Error-driven fixed-budget asr personalization for accented speakers," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7033–7037.

[12] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6124–6128.

[13] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," *arXiv preprint arXiv:1904.03288*, 2019.

[14] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection." *Journal of Machine Learning Research*, vol. 9, no. 12, 2008.

[15] T. Asami, R. Masumura, H. Masataki, M. Okamoto, and S. Sakauchi, "Training data selection for acoustic modeling via submodular optimization of joint kullback-leibler divergence," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] A. Sethy, P. G. Georgiou, B. Ramabhadran, and S. Narayanan, "An iterative relative entropy minimization-based data selection approach for n-gram model adaptation," *IEEE transactions on audio, speech, and language processing*, vol. 17, no. 1, pp. 13–23, 2009.

[17] A. Woodward, C. Bonnín, I. Masuda, D. Varas, E. Bou-Balust, and J. C. Riveiro, "Confidence measures in encoder-decoder models for speech recognition." in *INTERSPEECH*, 2020, pp. 611–615.