Accented Speech Recognition With Accent-specific Codebooks

Heting , Priyam, Xulin, Mahir, Kamila

2024/02/14

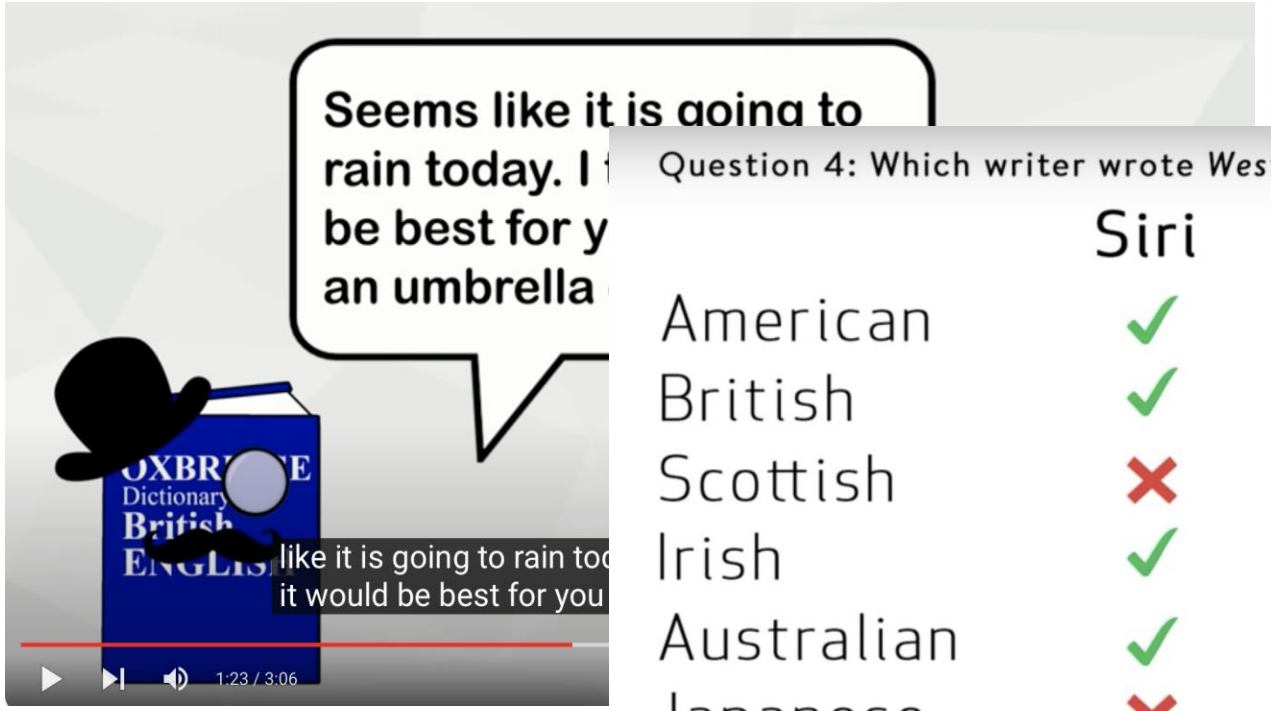
Overview

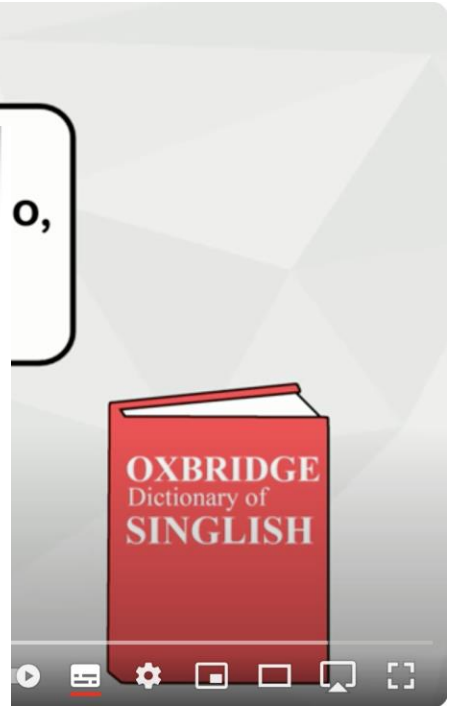
1. **Introduction - Kamila**
2. **Related Works - Kamila**
3. **Architecture - Priyam**
4. **Experiments – Xulin & Mahir**
5. **Discussion - Heting**
6. **Conclusion - Heting**

Introduction

British Accent

Singlish Accent

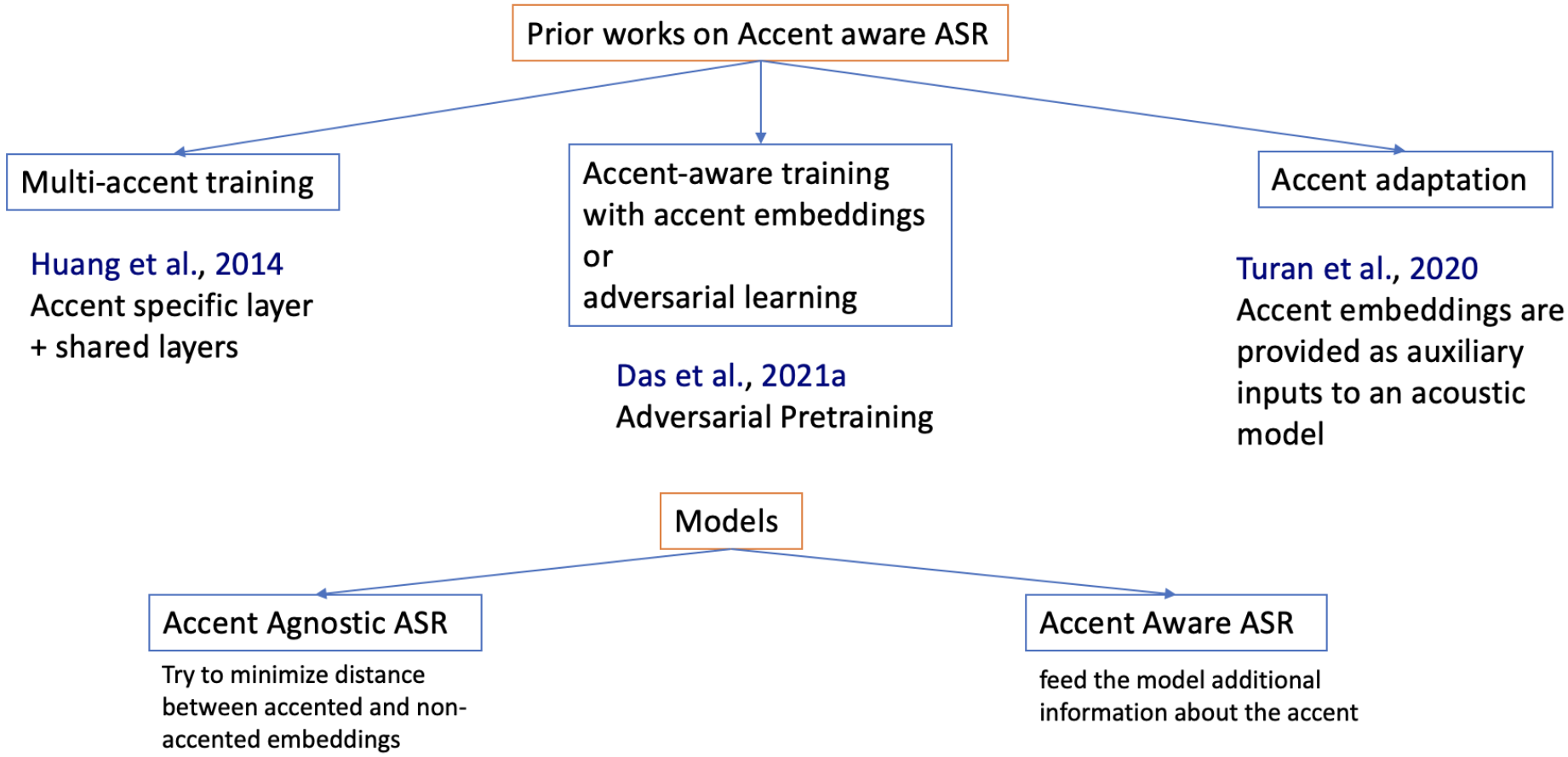




https://www.youtube.com/watch?v=1AnPurpl81c

Wired home assistants exper on in everyday life?
https://www.youtube.com/w S2VnDveP8

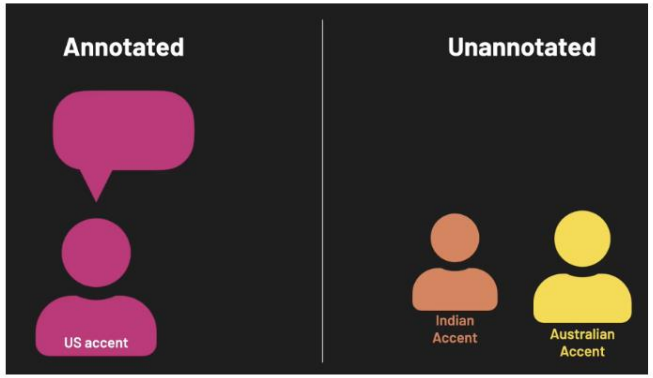
Related Works



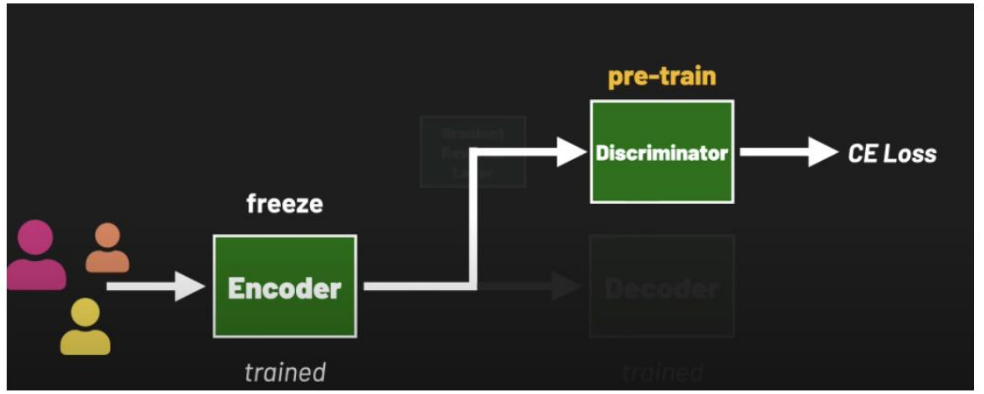
Main Difference:

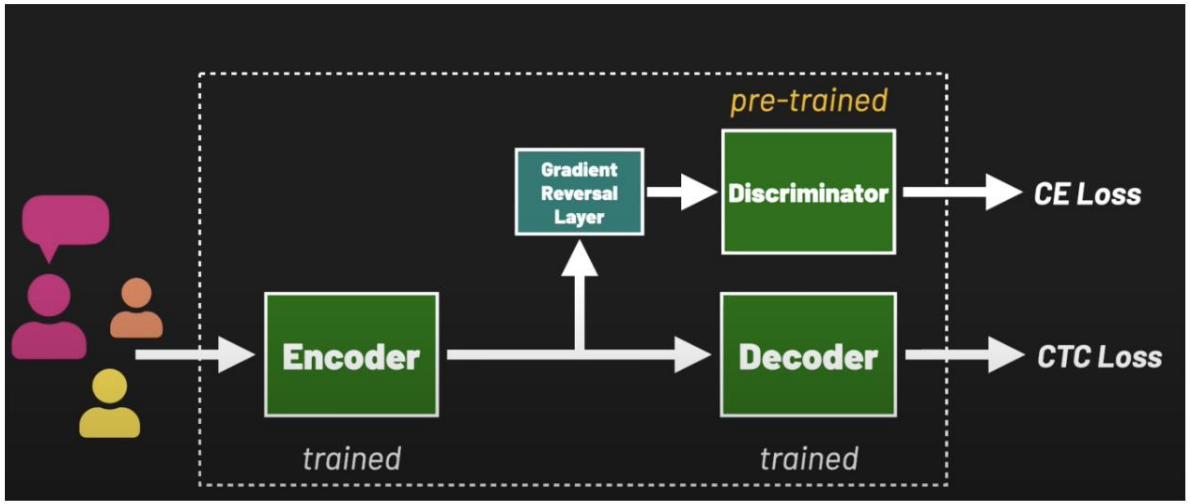
- End-to-end accent embeddings
- Beam-search decoding that searches over different accent combinations

Related Works



Pre-train on accent labels

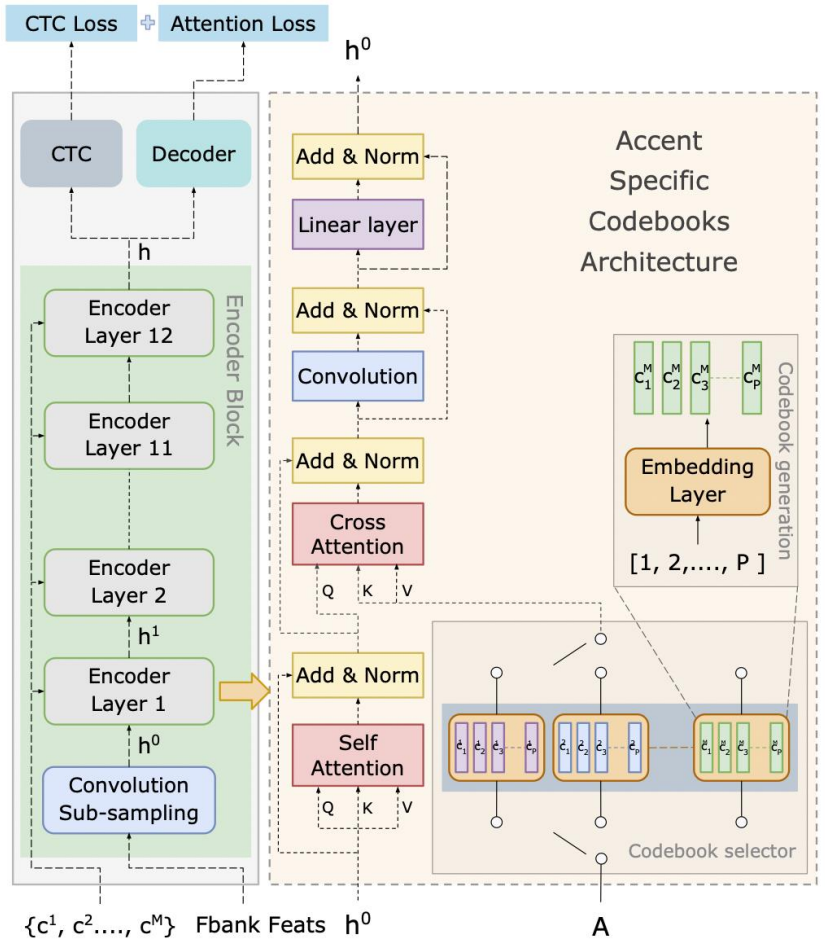




Best of Both Worlds: Robust Accented Speech Recognition with Adversarial Transfer Learning

By Nilaksh Das, Sravan Bodapati, Monica Sunkara, Sundararajan Srinivasan, Duen Horng Chau

Architecture Novelties



There were a few main changes made to the standard quantization method seen in other ASR models:

- Construction of codebooks that embed accent specific information
- Modified Beam-Search algorithm to handle inference when Accent Information is not provided.

Also cool to see they incorporated a joint CTC + Attention Loss! You can see more about this in *"Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning" (Kim et. al.)*

Codebook Construction and Training

```
if self.use_codebooks:
    self.no_codebooks = codebooks_per_accent
    self.no_accents = no_accents
    self.codebook_embedding = torch.nn.ModuleList([
        torch.nn.Embedding(
        num_embeddings = self.no_codebooks,
        embedding_dim = attention_dim
    ) for _ in range(no_accents)])
```

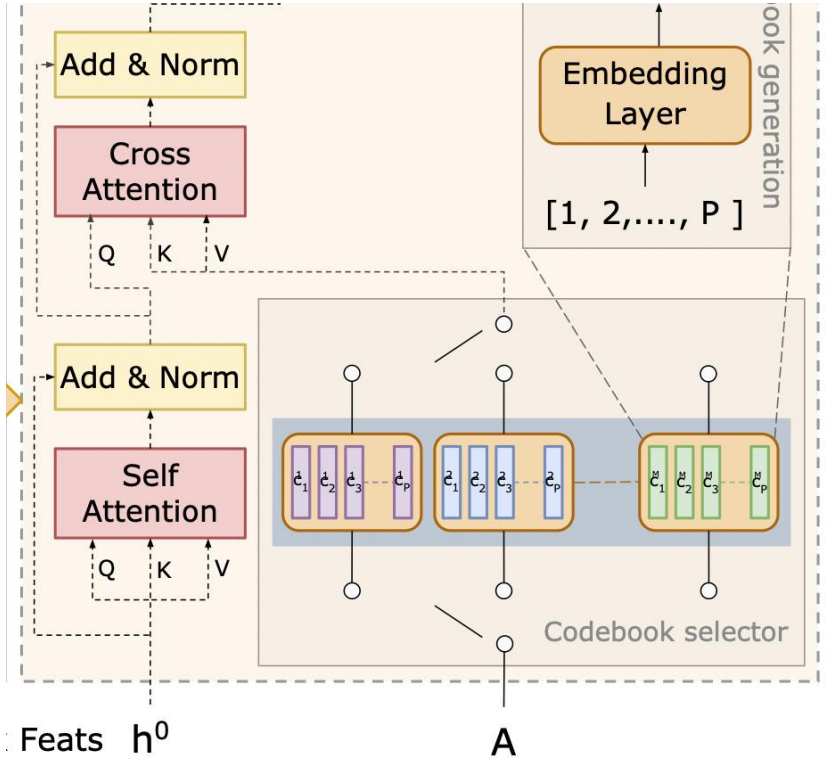
During training, we pass in our accent labels so we can index the specific codebook we want and encode our data

In the training corpus we can enumerate the number of accents that exist and generate a unique codebook for each one.

```
if self.use_codebooks:
    # Create codebooks for each data point.
    codebooks = []
    for label in accent_labels:
        arr = torch.tensor([i for i in range(self.no_codebooks)],dtype=torch.int).to(xs[0].device)
        codebooks.append(self.codebook_embedding[label](arr))

    codebooks = torch.stack(codebooks)
else:
    codebooks = None
```

Distributing Accent information Across Sequence



$$\{\alpha_{j,1}, \alpha_{j,2}, \ldots, \alpha_{j,P}\} =$$

$$\text{softmax}\left(\frac{(W_q^i \mathbf{H}_j)(W_k^i \mathbf{c})^T}{\sqrt{d}}\right)(W_v^i \cdot \mathbf{c})$$

Cross Attention modules were also incorporated to learn attention scores of how a sequence of audio attends over the codebook

```
if self.use_codebooks:
    if self.normalize_before:
        x = self.norm_codebook(x)

    residual = x
    codebook_context = self.codebook_attention(x,codebooks,codebooks,None)

    x = residual + stoch_layer_coeff * self.dropout(codebook_context)

    if not self.normalize_before:
        x = self.norm_codebook(x)
```

Modified Beam Search

Input : $\mathbf{x}$: speech input
$\mathcal{V}$: list of vocabulary tokens
$n_{\max}$: maximum hypothesis length
k: maximum beam width
y: output prediction so far
$\texttt{score}_A(.,.,.)$: scoring function

$\mathcal{B}_0 = \{\langle 0, \texttt{<sos>}, 1\rangle \ldots, \langle 0, \texttt{<sos>}, M\rangle\}$
for $t \in \{1, \ldots, n_{\max} - 1\}$ **do**
  $\mathcal{B} \leftarrow \phi$
  for $\langle s, y, A\rangle \in \mathcal{B}_{t-1}$ **do**
    if y.last() == <eos> **then**
      $\mathcal{B}$.add($\langle s, y, A\rangle$)
      continue
    for $v \in \mathcal{V}$ **do**
      $s \leftarrow \texttt{score}_A(\mathbf{x}, y \circ v, A)$
      $\mathcal{B}$.add($\langle s, y \circ v, A\rangle$)
  $\mathcal{B}_t = \mathcal{B}$.top($k$)
return $\mathcal{B}$

The Beam Search algorithm is a technique to find the (hopefully) best sequence of states from a list of state probabilities.

Typically Beam search combines the probability up the current timestep and the output prediction to predict what the next best states could be. The Modified Beam Search changes this slightly to add an extra tag *A* that also incorporates the most likely accent seen till this timestep as well.

Experiments – Main Results (Xulin)

Method	Aggregated			Seen Accents					Unseen Accents								
	All	Seen	Unseen	AUS	CAN	UK	SCT	US	AFR	HKG	IND	IRL	MAL	NWZ	PHL	SGP	WLS
Trans. (Dong et al., 2018)	22.7	17.3	28.0	18.1	17.8	19.7	18.5	16.3	25.9	32.0	35.4	25.3	36.2	23.8	31.5	38.8	21.0
Conf. (Gulati et al., 2020)	18.9	14.0	23.7	13.8	**15.0**	15.7	**13.4**	13.3	21.5	27.2	29.4	21.4	32.2	19.9	26.1	34.7	17.9
I-vector (Chen et al., 2015)	18.9	14.1	23.6	13.9	15.0	16.1	14.6	13.3	21.7	27.2	29.5	21.2	**31.7**	19.3	27.2	**33.8**	18.0
MTL (Jicheng et al., 2021)	18.9	14.1	23.7	14.7	15.1	16.1	13.7	13.2	21.8	28.1	**29.1**	21.5	33.0	19.4	26.5	34.2	18.1
DAT (Das et al., 2021b)	**18.7**	**14.0**	**23.4**	**13.3**	15.3	**15.7**	15.5	**13.1**	**21.1**	**27.0**	29.5	**21.1**	32.2	**19.2**	**26.0**	34.4	**17.9**
CA	**18.2**†	**13.6**	**22.9**	**11.5**	**14.8**	**14.9**	**9.7**	**13.1**	**21.0**	**25.7**	**29.1**	**20.7**	**30.9**	**18.5**	**25.8**	**33.7**	**17.9**

- **Model trained and tested on MCA-Accent-100**

- **Comparison with baselines:**

Network Architecture: Conformer > Transformer
Accent Augmentation (All w Conf): CA (Proposed) > DAT > I-vector ~= MTL

- **SOTA Performance:**

Proposed CA system outperforms other systems across all the seen and unseen accents

- **Configurations:**

50 codebook entries, incorporated into 12 Transformer Encoder layers

Experiments – Zero Shot (Xulin)

Method	All	Accents					
		ARA	HIN	KOR	MAN	SPA	VIA
Conformer	33.3	30.4	30.4	26.9	37.9	30.3	43.5
I-vector	33.6	31.0	31.2	27.2	38.0	30.4	43.9
MTL	33.4	30.4	30.6	26.9	38.7	30.1	43.7
DAT	33.5	30.7	30.8	26.8	38.3	30.1	43.9
CA	**32.6†**	**29.5**	**30.4**	**26.2**	**37.1**	**29.3**	**42.8**

- **Model trained on MCA-Accent-100, tested on unseen L2Arctic dataset**
- **Performance outperforms other baselines across all accents**
- **Ascertain the effectiveness of accent-specific codebooks**

Experiments – Data Size (Xulin)

Method	Overall	Seen	Unseen
Conf. (Gulati et al., 2020)	9.75	**6.04**	13.46
I-vector (Chen et al., 2015)	10.05	6.40	13.69
MTL (Jicheng et al., 2021)	10.02	6.33	13.70
DAT (Das et al., 2021b)	9.73	6.12	13.33
$\text{CA}_{L\in(1,\ldots,12)}(P=50)$	9.63	6.22	13.03
$\text{CA}_{L\in(1,\ldots,12)}(P=200)$	9.59	6.20	12.98
$\text{CA}_{L\in(1,\ldots,12)}(P=500)$	**9.55**	6.19	**12.92**

- **Comparison of models trained and tested on larger MCV-Accent(600 hours) data**
- **With larger dataset, proposed CA system still outperform baseline systems overall, and by a significant margin on unseen accents**

Experiments – Number of Parameters (Xulin)

Method	# of params	Overall	Seen	Unseen
Conf.	43M	18.87	14.05	23.67
Conf. w/ ↑ encoder units	46M	18.89	14.02	23.74
Conf. w/ ↑ attention dim	46M	18.77	14.02	23.51
$\mathrm{CA}_{L\in(1,\ldots,12)}(P=50)$	46M	**18.22**	**13.57**	**22.86**

- **Although proposed CA system outperforms other baselines, it has more parameters (46M vs 43M) which might be an important factor of final performance**
- **Experiments are done to discount the effect of parameter size**
- **Proposed CA models still outperform conformer with a significant performance gap with same number of parameters**

Experiments – Balanced Dataset (Xulin)

Method	Overall	Seen	Unseen
Conformer	19.30	14.73	23.86
$\mathrm{CA}_{L\in(1,\ldots,12)}(P=50)$	**18.88**	**14.61**	**23.13**

- **Comparison of models trained and tested on balanced MCA-Accent-100 Dataset**
- **Proposed method is still effective compared to conformer baseline with balanced data setting**

Experiments — Ablation Studies (Mahir)

Method	Overall	Seen	Unseen
$\mathrm{CA}_{L\in(1,\ldots,12)}(P=25)$	18.33	13.76	22.89
$\mathrm{CA}_{L\in(1,\ldots,12)}(P=50)$	**18.22**	**13.57**	**22.86**
$\mathrm{CA}_{L\in(1,\ldots,12)}(P=100)$	18.36	13.85	22.86
$\mathrm{CA}_{L\in(1,\ldots,12)}(P=200)$	18.41	13.69	23.12
$\mathrm{CA}_{L\in(1,\ldots,12)}(P=500)$	18.39	13.68	23.09
$\mathrm{CA}_{L\in(1,\ldots,4)}(P=50)$	**18.30**	13.95	**22.64**
$\mathrm{CA}_{L\in(1,\ldots,8)}(P=50)$	18.31	**13.86**	22.75
$\mathrm{CA}_{L\in(9,\ldots,12)}(P=50)$	18.92	14.24	23.59
$\mathrm{CA}_{L\in(5,\ldots,12)}(P=50)$	18.45	13.84	23.05
$\mathrm{CA}_{L\in(1,\ldots,12)}(P_{\mathrm{rand}}=50)$	18.30	13.65	22.95

- **Codebook size changes?**
  - Smaller codebook (expectedly) degrades performance
  - Larger codebooks overfit to seen accents
- **Cross-attention only at certain layers?**
  - More helpful when closer to the acoustics, since accent distinctions more prevalent
- **Random codebooks???**

Experiments — Ablation Studies (Mahir)

Method	Aggregated			Seen Accents					Unseen Accents								
	Overall	Seen	Unseen	AUS	CAN	UK	SCT	US	AFR	HKG	IND	IRL	MAL	NWZ	PHL	SGP	WLS
Transformer	22.68	17.34	28.01	18.11	17.81	19.73	18.50	16.31	25.86	31.97	35.39	25.25	36.25	23.83	31.50	38.78	21.04
Conformer (Base)	18.87	14.05	23.67	13.82	**15.02**	15.74	**13.36**	13.30	21.47	27.18	29.39	21.38	32.20	19.86	26.13	34.69	17.88
I-vector sum	18.87	14.15	23.58	13.88	15.02	16.07	14.62	13.31	21.66	27.18	29.53	21.18	**31.72**	19.33	27.22	**33.81**	17.98
Base + Classifier	18.91	14.12	23.69	14.73	15.10	16.08	13.72	13.19	21.83	28.13	**29.15**	21.46	32.97	19.38	26.51	34.24	18.08
DAT	**18.70**	**14.00**	**23.38**	**13.30**	15.30	**15.72**	15.52	**13.15**	**21.15**	**26.95**	29.53	**21.15**	32.16	**19.22**	**26.03**	34.43	**17.93**
$CA_{L\in(1,\dots,12)}(P=25)$	18.33	13.76	22.89	13.05	14.90	15.33	10.92	13.12	20.82	26.41	29.29	20.70	31.55	18.56	26.10	33.30	**16.58**
$CA_{L\in(1,\dots,12)}(P=50)$	**18.22**	**13.57**	22.86	**11.54**	**14.81**	14.91	**9.66**	13.15	20.95	25.66	29.15	20.72	30.87	**18.47**	**25.81**	33.68	17.92
$CA_{L\in(1,\dots,12)}(P=100)$	18.36	13.85	22.86	12.89	14.91	15.46	10.92	13.24	20.77	25.89	28.87	**20.41**	32.44	18.76	26.24	**32.93**	17.77
$CA_{L\in(1,\dots,12)}(P=200)$	18.41	13.69	23.12	13.00	14.81	15.06	11.10	13.12	21.57	26.78	**28.30**	20.93	30.95	18.97	25.97	33.17	17.88
$CA_{L\in(1,\dots,12)}(P=500)$	18.39	13.68	23.09	12.04	14.93	15.40	11.10	**13.05**	21.22	26.52	28.77	20.57	33.13	18.78	25.97	33.81	18.39
$CA_{L\in(1,\dots,4)}(P=50)$	18.30	13.95	**22.64**	13.22	15.53	15.49	10.74	13.24	20.79	26.06	28.58	20.52	31.43	17.87	26.27	32.98	17.72
$CA_{L\in(1,\dots,8)}(P=50)$	18.31	13.86	22.75	13.14	15.07	15.76	10.56	13.12	**20.50**	**25.52**	28.70	21.03	**30.70**	18.53	25.59	33.10	18.03
$CA_{L\in(9,\dots,12)}(P=50)$	18.92	14.24	23.59	13.30	15.36	15.53	13.27	13.67	21.50	27.18	28.89	21.61	32.52	18.98	26.64	34.43	19.59
$CA_{L\in(5,\dots,12)}(P=50)$	18.45	13.84	23.05	12.37	15.43	15.42	10.92	13.18	21.08	26.75	28.40	20.60	31.76	18.68	26.52	34.00	18.13
$CA_{L\in(1,\dots,12)}(P_{\text{rand}}=50)$	18.30	13.65	22.95	12.34	15.19	**14.89**	11.64	13.06	20.77	25.57	29.48	20.98	31.88	18.62	25.87	33.28	17.82

Method	Aggregated			Seen Accents					Unseen Accents								
	Overall	Seen	Unseen	AUS	CAN	UK	SCT	US	AFR	HKG	IND	IRL	MAL	NWZ	PHL	SGP	WLS
Conformer (Base)	9.75	**6.04**	13.46	4.95	**6.81**	7.15	4.42	**5.64**	11.81	16.50	14.90	**12.61**	20.43	10.49	**15.74**	21.26	**7.98**
I-vector	10.05	6.40	13.69	4.67	7.53	7.43	4.06	6.04	12.05	17.45	14.76	13.08	20.31	10.57	15.82	**21.22**	8.91
MTL	10.02	6.33	13.70	5.30	7.59	7.39	**3.97**	5.86	12.19	16.30	**14.25**	13.23	19.58	10.64	16.30	21.54	8.50
DAT	**9.73**	6.12	**13.33**	**4.56**	7.50	**6.87**	4.96	5.74	**11.76**	**16.19**	14.62	12.96	**18.97**	**9.91**	15.84	21.50	8.13
$CA_{L\in(1,\dots,12)}(P=50)$	9.63	6.22	13.03	4.67	**7.36**	7.11	3.07	**5.90**	11.63	15.64	14.06	12.48	18.97	**9.73**	15.60	21.05	8.29
$CA_{L\in(1,\dots,12)}(P=200)$	9.59	6.20	12.98	4.92	7.47	6.83	3.52	5.91	**11.47**	15.96	**13.87**	12.25	19.30	9.98	**15.17**	20.90	**8.08**
$CA_{L\in(1,\dots,12)}(P=500)$	**9.55**	**6.19**	**12.92**	**4.40**	7.60	**6.67**	**2.62**	5.99	11.57	**15.18**	14.13	**12.20**	**17.84**	10.00	15.34	**20.71**	8.39

Experiments — Single Accent Inference (Mahir)

Accent used	Seen Accents					Unseen Accents								
	AUS	CAN	UK	SCT	US	NWZ	IRL	AFR	MAL	WLS	HKG	IND	PHL	SGP
Australia	**11.5**	19.5	17.0	18.1	17.4	**18.7**	24.3	22.0	33.7	21.1	29.8	32.5	30.1	37.8
Canada	20.5	**14.7**	20.0	15.7	13.5	25.7	**21.4**	24.5	32.7	21.8	27.4	29.6	26.6	35.4
England	13.8	17.7	**15.0**	14.4	16.2	21.1	22.0	**21.5**	**32.3**	**18.0**	27.0	29.9	27.1	34.8
Scotland	20.7	17.8	19.1	**10.2**	16.4	25.7	22.6	24.4	34.4	21.3	28.2	33.5	29.0	36.6
US	20.2	**14.7**	19.4	15.5	**13.2**	24.7	21.7	23.4	32.4	22.2	**27.0**	**28.1**	**25.8**	**34.3**

- **Using only one accent's codebook for decoding?**
  - **Lower WERs when using geographically proximate accent codebooks for unseen accents**
  - **(some personal ideas below regarding some of the 'proximities'):**
    - **Irish ancestry being the 2nd largest European subgroup in Canada?**
    - **Increased consumption of American media throughout former British colonies in Asia? (note '32.4' vs. '32.3' in the Malaysia column, '27.0' vs. '27.0' in the Hong Kong column)**
  - **Errors still much higher than when beam-searching**

Experiments — Variants in Beam-Search Decoding (Mahir)

- **Beam-search variants at inference time?**
  - **Standard beam search? Not great**
  - **One k-width beam search per codebook? Better than what was proposed, but takes too long**
  - **One k/M-width beam search per codebook? More efficient but under-utilizes beam slots**

Method	All	Seen	Unseen	Inference Time
$\mathbb{B}_0$: Standard beam search	18.87	14.05	23.67	1.0
$\mathbb{B}_1$: M full beam searches	**18.10**	**13.48**	**22.71**	5.02
$\mathbb{B}_2$: M split beam searches	18.30	13.61	22.97	1.14
$\mathbb{B}_3$: Joint beam search	18.22	13.57	22.86	1.16

Discussion and Analysis - Codebook Utilization (Heting)

- Figure 2: Across seen accents, a diagonal dominance
  - Evident for Australia, England and Scotland accents
  - US and Canada have examples evenly divided among each other
- Figure 2: Among unseen accents
  - Australia-specific codebook is picked up most by New Zealand test utterances

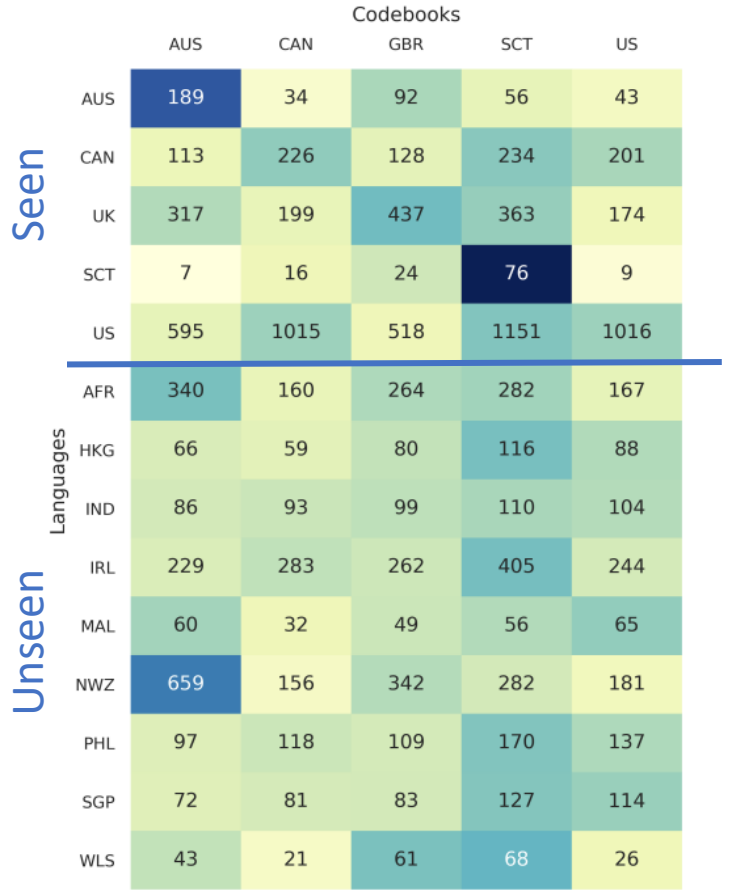


Figure 2: Heatmap showing which codebooks are chosen during inference across seen and unseen accents. For example, the third cell in the first row shows that 92 out of 413 Australian-accented utterances used the codebook belonging to England during decoding.

Discussion and Analysis - Active Accents during Joint Beam-search (Heting)

- Figure3: Entropy across during beam-search decoding step
  - Compute the distribution of samples in the beam across the five seen accents
  - Plot the average entropy of this distribution across all test instances
  - four to five seen accents are active until time-step 20, after which certain accents gain more prominence
- Figure 4: Probabilities across seen accents in the beam for a single Wales accented test sample
  - All accents are active at the start of the utterance,
  - England becomes the dominant accent towards the end

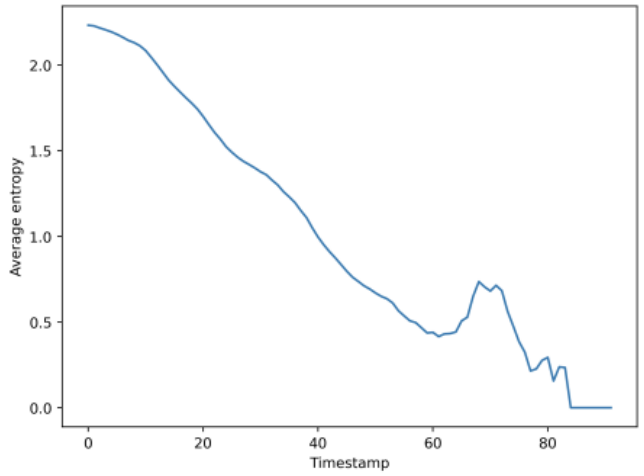


Figure 3: Progression of average test entropy of the probability distribution across seen accents.

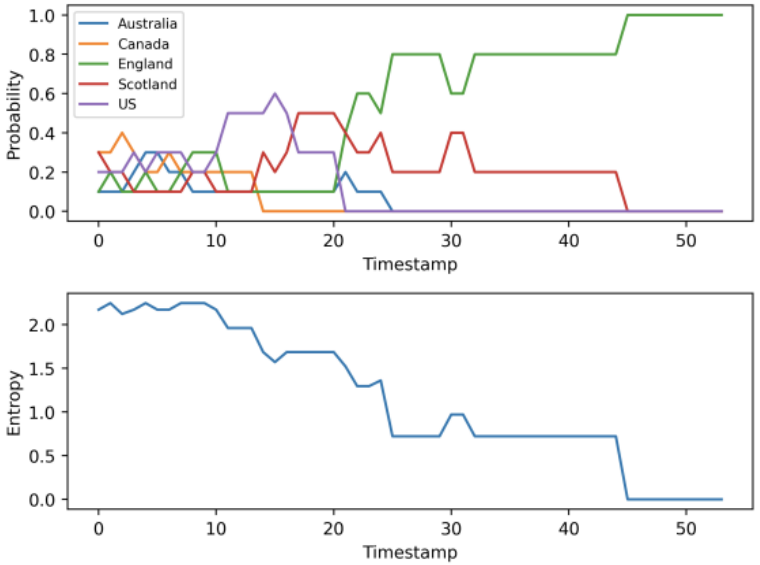


Figure 4: Progression of the probability/entropy across seen accents for a single Wales-accented test sample.

Discussion and Analysis - Alternatives to Joint Beam-search (Heting)

- Two alternatives to joint beam-search
  - Jointly trained an accent classifier with ASR.
    - During inference, this classifier provides pseudo-accent labels across seen accents to choose the codebook
  - Adds a learnable gate to each codebook entry
    - Instead of picking a fixed subset of codebook entries
    - Train learnable gate at each encoder layer jointly with ASR to pick a designated codebook entry corresponding to the underlying accent of the utterance
    - During inference, the learned gates determine the codebook entries to be used for each encoder layer
- Both performed better than the Conformer baseline but were equivalent to the DAT approach
  - Maybe due to the lack of a strong accent classifier
  - Lack of appropriate learning in the gates to capture accent information

Discussion and Analysis - Why Performance Improvements on Unseen Accents (Heting)

- The model is designed to choose (seen) accent codebooks that best fit the underlying (unseen) accent.
  - Analogous to how humans use familiar accents to tackle unfamiliar ones
  - During inference, the model searches through seen accent codebooks and chooses entries that are most like the unseen accents in the test instances

Limitations (Heting)

- The codebook size is a hyperparameter that needs to be finetuned for each task.
- Currently employ accent-specific codebooks, one for each accent.
  - This does not scale very well and does not enable sharing of codebook entries across accent codebooks.
  - Instead, we could use a single (large) codebook and use learnable gates to pick a subset of codebook entries corresponding to the underlying accent of the utterance.
- The proposed joint beam-search leads to a 16% increase in computation time at inference.
- The joint beam-search allows for each utterance at test-time to commit to a single seen accent.
  - Parts of an utterance might benefit from one seen accent, while other parts of the same utterance might benefit from a different seen accent.
  - Such a mix-and-match across seen accents is currently not part of the proposed approach.
  - Accommodating for such effects might improve the model further

Conclusion (Heting)

- Propose a new end-to-end technique for accented ASR
  - Uses accent-specific codebooks and cross-attention to achieve significant performance improvements on seen and unseen accents at test time.
- Experiment with the Mozilla Common Voice corpus and show detailed ablations over our design choices.
- Empirically analyze whether our codebooks encode information relevant to accents.
  - The effective use of codebooks for accents opens up future avenues to encode non-semantic cues in speech that affect ASR performance, such as types of noise, dialects, emotion styles of speech, etc.