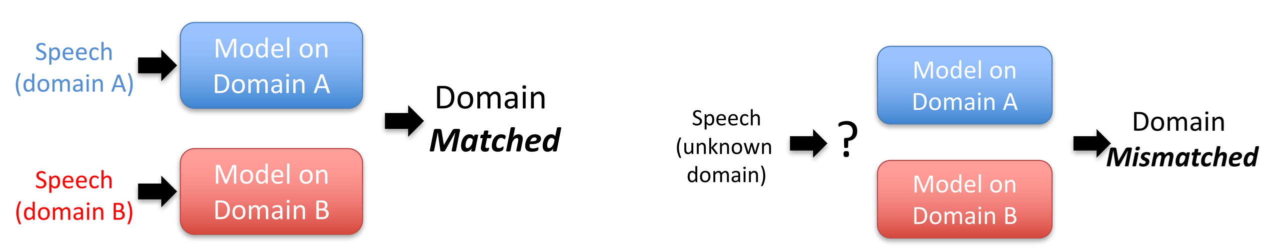
Domain Attentive Fusion for End-to-end Dialect Identification with Unknown Target domain

Suwon Shon[1], Ahmed Ali[2], James Glass[1]
MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA, USA[1]
Qatar Computing Research Institute, HBKU, Doha, Qatar[2]

Motivation

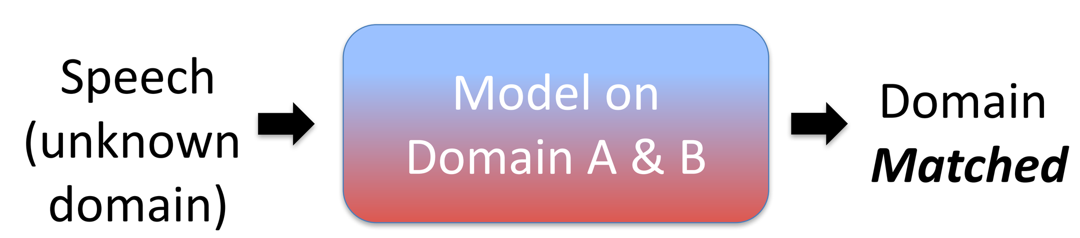
- Domain mismatch issue on dialect identification task
-> one of the challenges in real-world spoken language processing
- Ideally we have domain matched result if we have enough data on multiple domains and know about the input data
- Proposed domain attentive fusion network to deal with the unknown target domain

➢ **Unknown target domain on end-to-end systems**



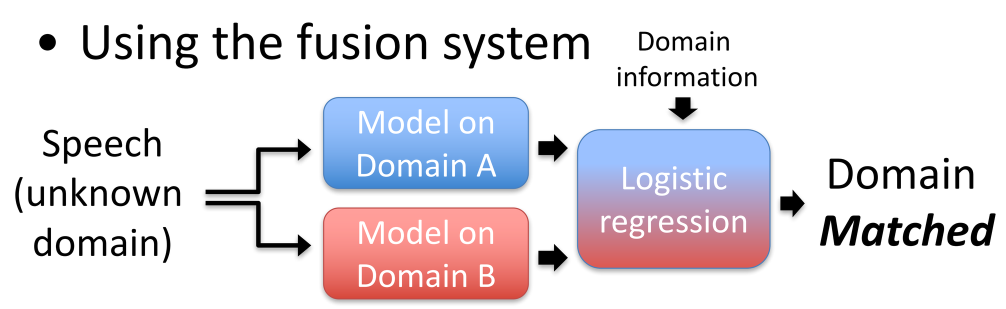
Known target domain *unknown target domain*

➢ **Current solution**

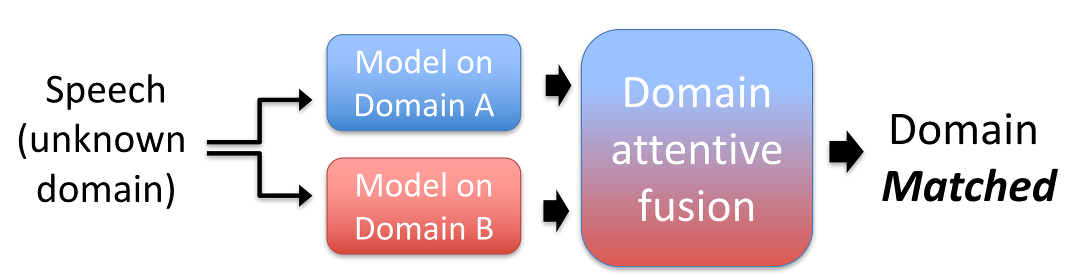


Training multiple domain system
-> does not show best performance on each domain

- Proposed system
  - Using the fusion system



Fusion of multiple systems
-> Still need prior



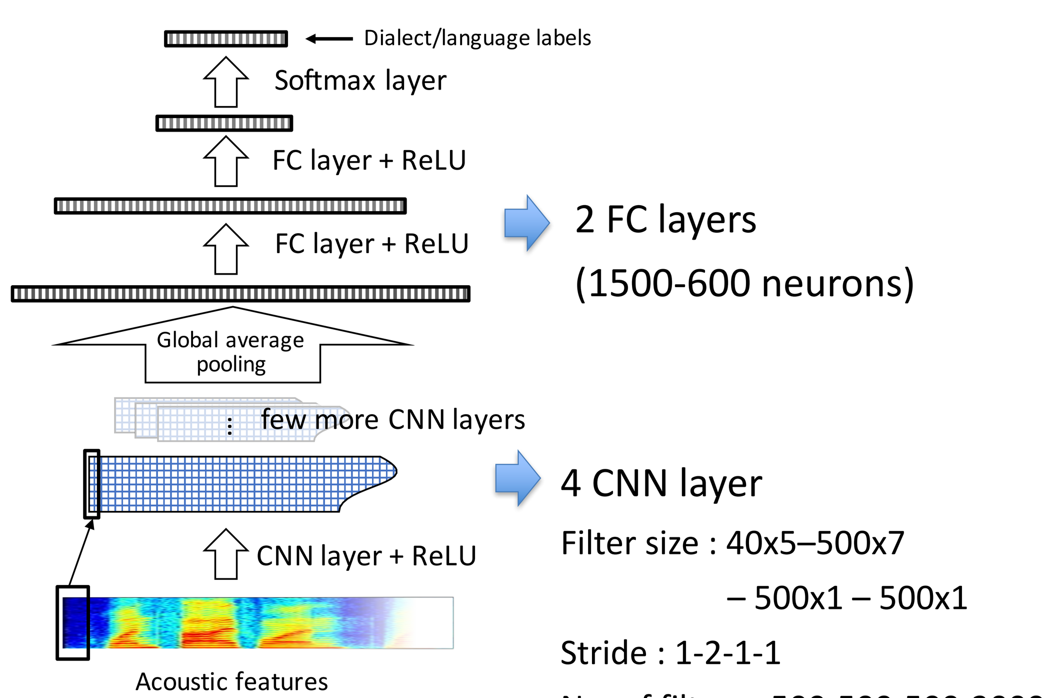
Domain attentive fusion systems
*-> proposed a self-attention based fusion approach when the input comes from an **unknown domain***

Dataset Collection

- Identifying 5 Dialects
  - Modern Standard Arabic, Egyptian Levantine, Gulf, North African
- Multiple Domains
  - MGB-3 : Collected from broadcast system
    - 53.6 hours (recorded), 10.0 hours(high qual.)
  - VarDial 2018 : Collected from YouTube
    - 1000 hours

End-to-end Dialect ID

- CNN based End-to-end neural network structure



Network structure

- Performance based on the training data
  - Using both domains for training performs generally better for both domain
  - Logistic regression based fusion with an individually trained network on a single domain gives better performance than training with multiple domains
  - However, fusion gives the best performance when the input domain is known a priori

	Training data	System ID	DID Accuracy (%)	
			MGB-3 Test	VarDial 2018 Test
Single domain	MGB-3 Train + MGB-3 Dev	$\mathcal{A}$	**65.82**	48.87
Single domain	VarDial 2018 Train	$\mathcal{B}$	51.27	**86.40**
Multi domain	MGB-3 Train + MGB-3 Dev + VarDial 2018 Train	$\mathcal{A}+\mathcal{B}$	61.86	81.53
	Fusion of $\mathcal{A}$ and $\mathcal{B}$ (optimized for $\mathcal{A}$)	-	**68.63**	77.57
	Fusion of $\mathcal{A}$ and $\mathcal{B}$ (optimized for $\mathcal{B}$)	-	57.84	**86.94**

Baseline performance and its fusion

Experimental results

	Training data	MGB-3 Test Acc.	MGB-3 Test EER	MGB-3 Test Cavg	VarDial 2018 Test Acc.	VarDial 2018 Test EER	VarDial 2018 Test Cavg	Averaged Acc.	Averaged EER	Averaged Cavg
Single domain	MGB-3 Train + MGB-3 Dev ($\mathcal{A}$)	65.82	20.43	19.60	48.87	28.39	28.50	58.35	24.41	24.05
Single domain	VarDial 2018 Train ($\mathcal{B}$)	51.27	28.37	27.41	86.40	9.57	9.96	68.84	18.97	18.69
Multi domain	MGB-3 Train + MGB-3 Dev + VarDial 2018 Train ($\mathcal{A}$+$\mathcal{B}$)	61.86	22.92	21.41	81.53	11.13	11.76	71.70	17.03	**16.59**
	Logistic regression fusion of $\mathcal{A}$ and $\mathcal{B}$ (optimized for $\mathcal{A}$)	**68.63**	**19.05**	**18.04**	77.57	13.78	14.16	**73.10**	**16.42**	16.10
	Logistic regression fusion of $\mathcal{A}$ and $\mathcal{B}$ (optimized for $\mathcal{B}$)	57.84	24.36	23.35	**86.94**	**9.23**	**9.56**	72.39	16.80	16.46
	Using fusion layer on $\mathcal{A}$ and $\mathcal{B}$ (Figure 1)	67.69	19.30	18.39	82.86	11.19	11.58	75.28	15.25	14.99
	Domain Attentive fusion of $\mathcal{A}$ and $\mathcal{B}$ (Figure 2 (a))	67.49	18.52	18.01	83.93	10.03	10.22	75.71	14.28	14.12
	Domain Attentive fusion of $\mathcal{A}$ and $\mathcal{B}$ (Figure 2 (b))	**68.23**	**18.30**	**17.69**	**85.01**	**9.13**	**9.40**	**76.62**	**13.72**	**13.55**

(Test on: MGB-3 Test, VarDial 2018 Test, Averaged)

Performance evaluation on multiple domains

- The proposed approach was verified in conditions where the test set domain was seen and unseen when training a network
- The traditional approach shows reasonable performance *only if the input domain is known a priori*
- Neural network based fusion approaches learn how to fuse networks from the training set and ***automatically calculate the weight of the network to contribute optimally for the random test input***
- The proposed approach performs consistently better on inputs from unknown domains compared to the traditional fusion approach
  - Improved **16%** in EER for seen domain input
  - Improved **4%** in EER for unseen domain input

Domain Attentive Fusion

- Self attention based fusion
  - Learning scalar score e_d for the language ID output $\mathbf{h_d}$ where $d \in \{D_1, D_2\}$ as

$$e_d = f(\mathbf{o}_d).$$

  - Scoring function

$$f(\mathbf{o}_d) = \mathbf{v}_d^T \tanh(\mathbf{W}_d \mathbf{o}_d + \mathbf{b}_d)$$

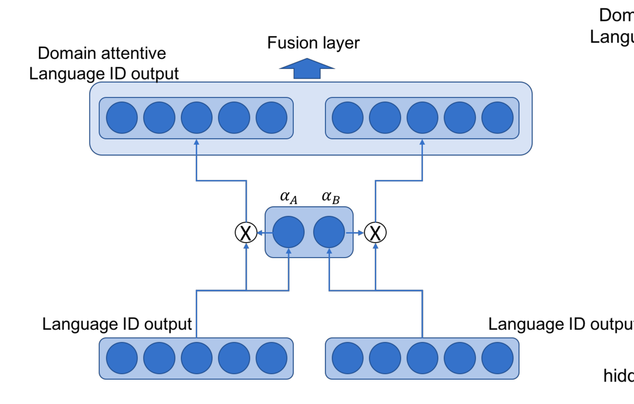
  - Normalized weights α_d is

$$\alpha_d = \frac{exp(e_d)}{(exp(e_A)+exp(e_B))}$$

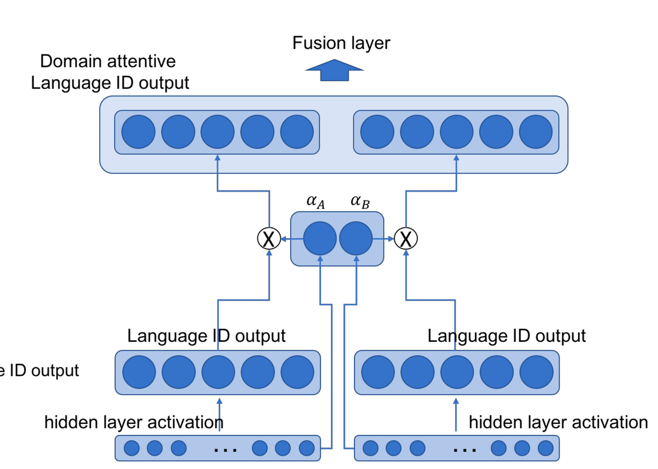
  - Domain attentive output is

$$\mathbf{o} = [\alpha_{D_1} * \mathbf{o}_{D_1}, \alpha_{D_2} * \mathbf{o}_{D_2}]$$

- Fusion structures
  1. Logistic regression
  2. Adding fully connected layer
  3. Domain attentive fusion using output
  4. Domain attentive fusion using hidden layer activation



Domain attentive layer using outputs



Domain attentive layer using hidden layer activation

Conclusion

- The traditional fusion strategy performs optimally when the input domain is known
- Training single network with multiple domain does not show best result on each domain
- We propose a neural network based self-attention fusion approach
- A domain attentive layer automatically decides the weights of multiple dialect ID systems by looking at the output or the embedding layer of each system
- The performance on each test set from different domains is even better than the single domain optimized fusion approach

Acknowledgements: This research was carried out in collaboration between the HBKU Qatar Computing Research Institute (QCRI) and the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL).