Attentive Temporal Pooling for Conformer-based Language Identification

Authors: Quan Wang, Yang Yu, Jason Pelecanos,
Yiling Huang, Ignacio Lopez Moreno

Date: 2022/06/30

ODYSSEY
June 28th-July 1st 2022, Beijing, China

Agenda

01 Introduction

02 Conformer-based lang-id

03 Attentive temporal pooling

04 Domain adaptation

05 Experiments

06 Conclusion

Section 1

Introduction

Introduction

- Problem:
  - Improve **language identification** accuracy across **multiple domains** for commercial applications
  - Special focus on increasing its robustness on **long-form speech**

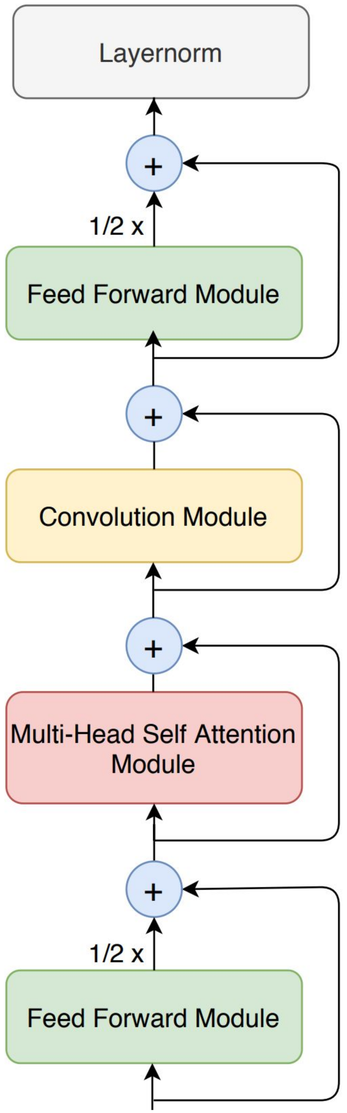
- Approach:
  - **Conformer-based** encoder structure
  - **Adaptive temporal pooling** via a recurrent form
  - Discriminatively trained output transform for **domain adaptation**

Section 2

Comformer-based lang-id

Conformer-based lang-id

- Conformer*:
  - Combining **convolutional neural networks** (CNN) and **transformers.**
  - Demonstrated performance improvements on ASR and other speech related tasks.
  - Drawback: limited context when training with short utterances.



Conformer block

* Gulati *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition", 2020

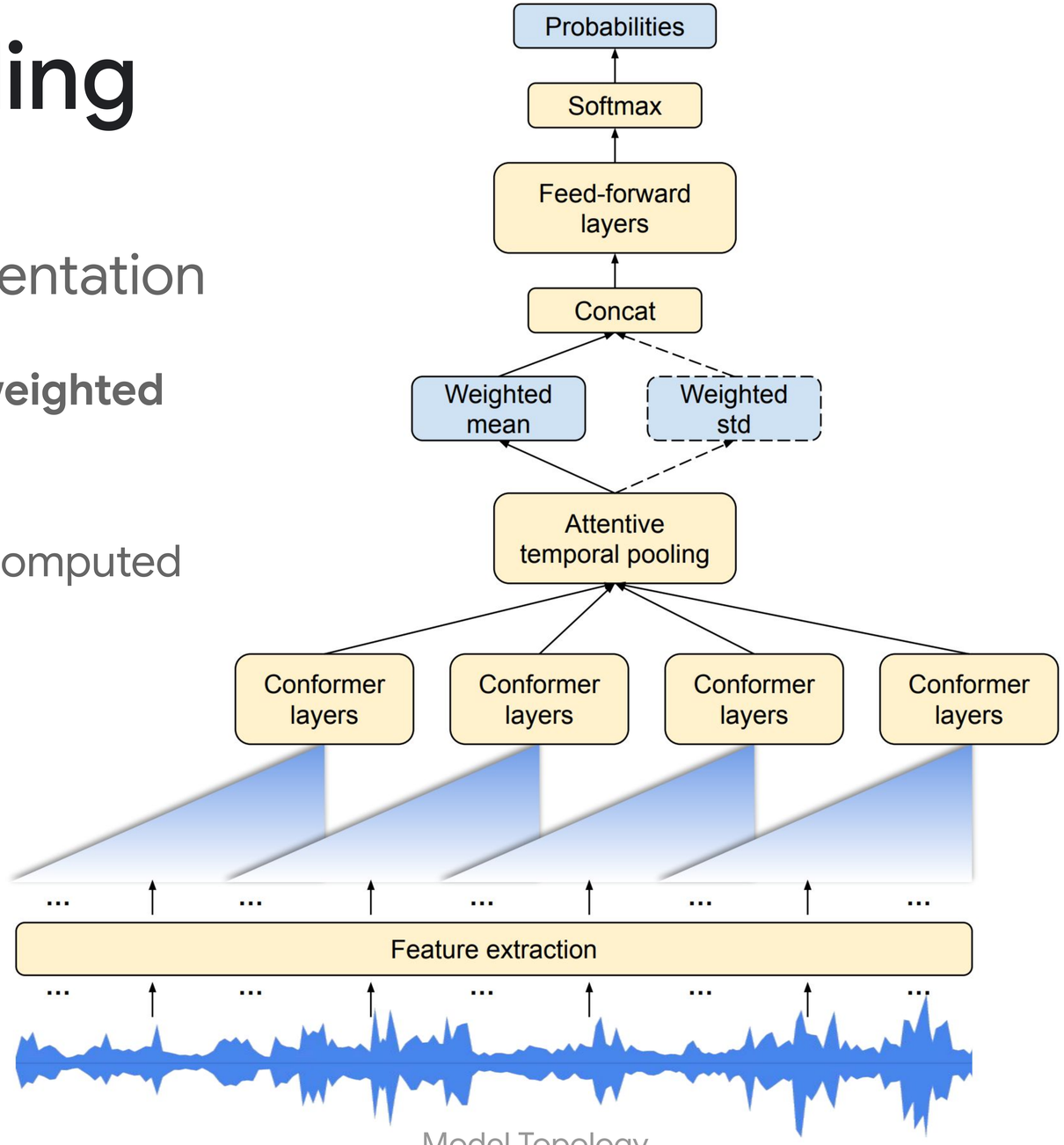
Section 3

Attentive temporal pooling

Attentive temporal pooling

- A weighted moving average implementation
  - Pooling based on **weighted mean** and **weighted standard deviation**
  - For each frame, **an attention weight** is computed based on the current embedding:

$$w_t = f_{\text{att}}(\mathbf{h}_t) + \epsilon$$



Model Topology

Attentive temporal pooling

- Tracking the following statistics accumulated with time:
  - Sum of weights: $\eta_t = \eta_{t-1} + w_t$
  - Weighted sum of outputs: $\boldsymbol{A}_t = \boldsymbol{A}_{t-1} + w_t \mathbf{h}_t$
  - Weighted sum of square of outputs: $\boldsymbol{Q}_t = \boldsymbol{Q}_{t-1} + w_t \mathbf{h}_t^2$

- Weighted mean and weighted standard deviation can be calculated using the sufficient statistics above:

$$\boldsymbol{\mu}_t = \frac{\boldsymbol{A}_t}{\eta_t}$$

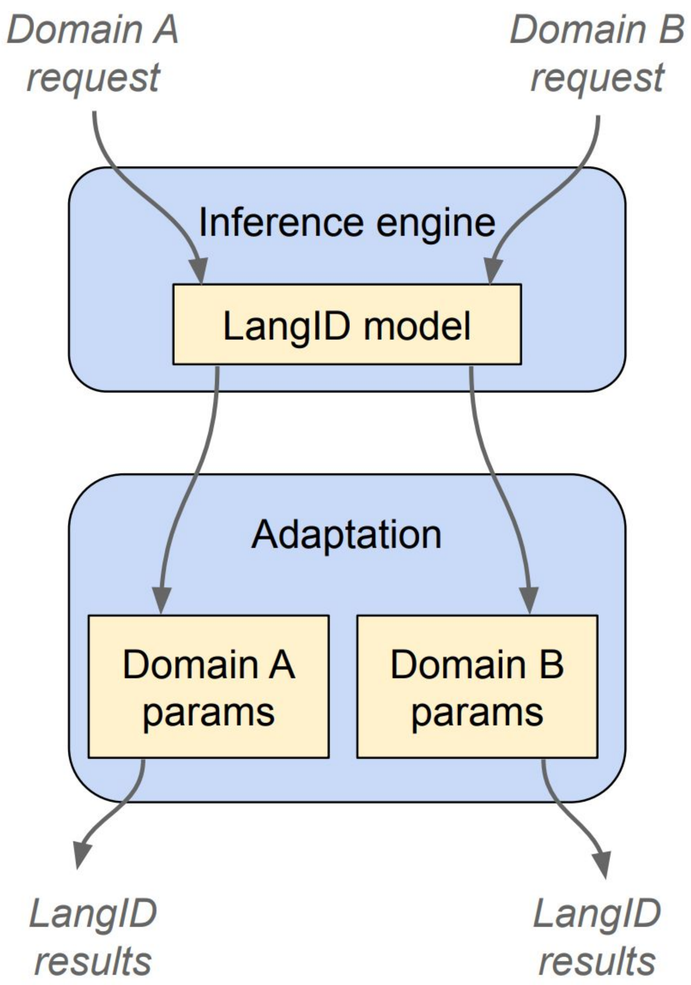
$$\boldsymbol{\sigma}_t = \sqrt{\frac{\boldsymbol{Q}_t}{\eta_t} - \boldsymbol{\mu}_t^2}$$

Section 4

Domain adaptation

Domain adaptation

- One model for all applications
- The domain parameters are used to adjust the output probabilities for each application
- We share two approaches:
  - Prior replacement: application specific prior probability updates
    - based on recording counts (not acoustics)
  - Discriminatively trained output transform
    - needs the recordings themselves



Adaptation to Domain A and B

Domain adaptation

- Approach 1: Prior replacement
  - Estimate the posterior probabilities of the languages using updated prior probabilities from the new application*
    - First, estimate the language prior probabilities based on the language distribution from samples:

$$P(L_i|D_{new}) = \frac{c_i + R}{\sum_{j=1}^{K}(c_j + R)}$$

    - Then, calculate the posterior probabilities:

$$P(L_i|X, D_{new}) = \frac{P(L_i|D_{new})P(L_i|X, D_{old})}{\sum_{j=1}^{K} P(L_j|D_{new})P(L_j|X, D_{old})}$$

* Bailer-Jones *et al.*, "Combining probabilities", 2011

Domain adaptation

- Approach 2: **Discriminatively** trained output transform
  - Transform predicted probability into a new probability distribution

$$\widetilde{\mathbf{p}} = \text{softmax}(\mathbf{a} \odot \log \mathbf{p} + \mathbf{b})$$

  - Minimize a regularized cross entropy between the new distribution with the ground truth language distribution on the dev-set

$$\arg\min_{\mathbf{a},\mathbf{b}} \left(\frac{1}{N} \sum_{i=1}^{N} L_{\text{cent}}(\widetilde{\mathbf{p}}^{(i)}, \mathbf{y}^{(i)}) + w_{\text{reg}}(||\mathbf{a} - \mathbf{1}|| + ||\mathbf{b}||) \right)$$

Section 5

Experiments

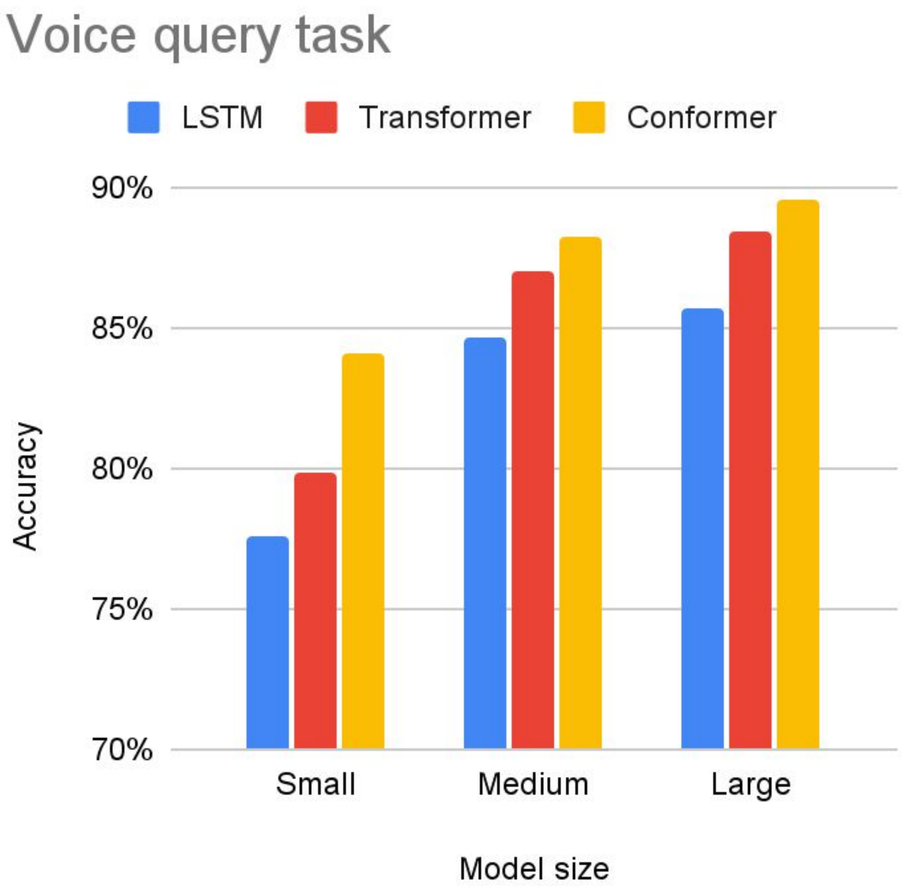
Experiment setup

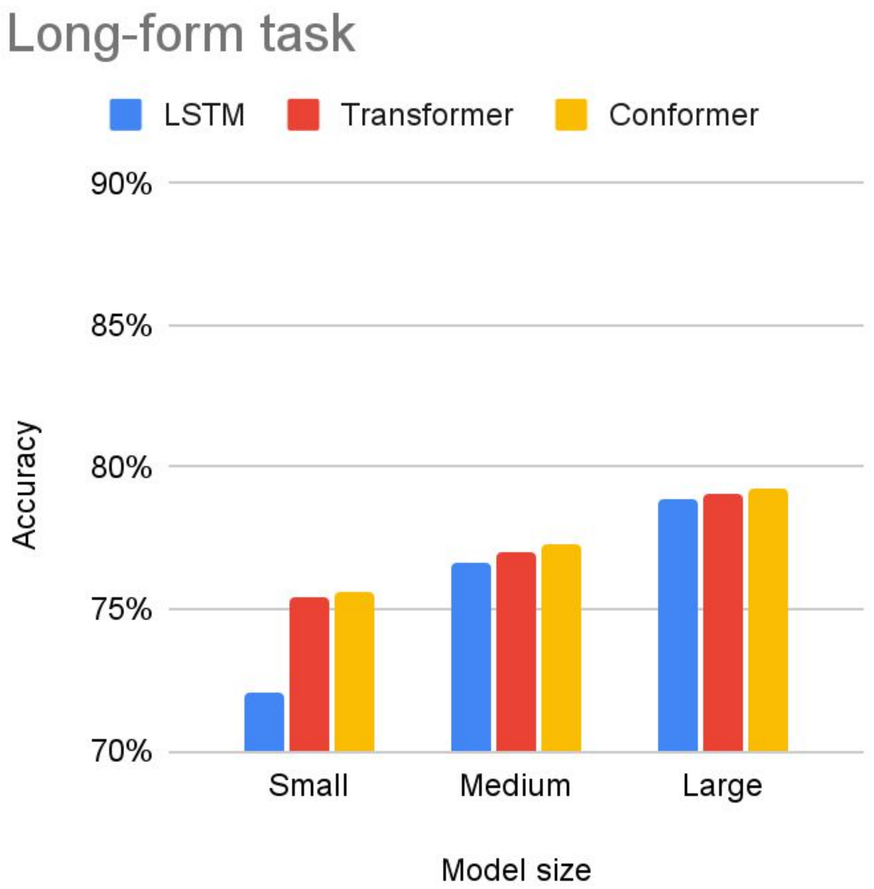
- Datasets:
  - Voice queries from Google Assistant:
    - Audio length: 3.3 ± 1.5s (std.)
  - Long-form utterances from YouTube:
    - Audio length: 20.7 ± 10.6mins (std.)
  - Training: 1M to 20M utterances per language for 65 languages
  - Evaluation: 20K utterances per language per domain

- Training frontend:
  - Log Mel-filterbank energy features
  - Data augmentation with multi-style training (MTR) and SpecAugment
  - A pre-trained Voice Activity Detector (VAD) to remove the non-speech parts

Experimental results

- Comparison of LSTM, transformer and conformer models
  - Small model: ~ 7M params, Medium model: ~ 30M params, Large model: ~120M params





Experimental results

- Comparison of different temporal pooling approaches

Model	Voice query avg. accu.	Long-form avg. accu.
Medium-size conformer	88.24%	77.26%
↳ + Mean pooling	88.18%	77.28%
↳ + Mean&std pooling	88.32%	77.35%
↳ + Weighted mean pooling	**88.92%**	77.45%
↳ + Weighted mean&std pooling	88.74%	**77.81%**

Experimental results

- Comparison of domain adaptation methods

Method	Domain adapted to	Voice query total accuracy (Perplexity: $PP = 33.8$)	Long-form total accuracy (Perplexity: $PP = 56.2$)
No adaptation	-	90.85%	77.94%
Prior replacement ($R = 0$)	Voice query	**91.76%**	54.21%
	Long-form	83.09%	**78.18%**
Prior replacement ($R = 4$)	Voice query	**91.74%**	74.67%
	Long-form	90.16%	**78.17%**
Output transform	Voice query	**92.32%**	76.18%
	Long-form	76.97%	**82.55%**

Section 6

Conclusions

Conclusions

- Presented a language identification system focusing on long-form and multi-domain performance.

- Compared different methods for several key components of lang-id
  - **Conformer** outperforms LSTM and Transformer as an encoder
  - The **attentive temporal pooling** helps compared with other pooling methods
  - The **discriminative output transform** shows better performance than **prior replacement**

Questions?

Appendix

Domain adaptation

- Relation between the prior replacement and discriminatively trained output transform
  - Fix the variable **a** as a vector of 1s:

$$\begin{aligned} & \mathbf{a} \odot log\mathbf{p} + \mathbf{b} \\ = & \mathbf{1} \odot log\mathbf{p} + log(e^{\mathbf{b}}) \\ = & log(\mathbf{p} \odot e^{\mathbf{b}}) \end{aligned}$$

  - Optimizing **b** is the same as optimizing prior probabilities