Transformer versus LSTM Language Models Trained on Uncertain ASR Hypotheses in Limited Data Scenarios

Imran Sheikh, Emmanuel Vincent, Irina illina*
Université de Lorraine, CNRS, Inria, Loria F-54000 Nancy, France
*irina.illina@loria.fr

Background

- Domain/Application specific ASR LMs require:
  - in-domain text, and/or
  - manually verified in-domain speech transcriptions
- Such domain specific text resources are scarce
- In-domain speech data also limited, e.g., in
  - the early development stages of a new application
  - in privacy-critical applications
  - for under-resourced languages

Goal: LMs from a limited amount (25–50h) of in-domain speech.
Problem: Training neural LMs on ASR decoded graphs.

Prior Art

- n-gram LMs on ASR lattices [*Kuznetsov et al., 2016*]
- Adaptation of RNN LMs on 1-best hypotheses [*Li et. al. 2018*]
- Train LSTM LMs on confusion networks [*Sheikh et. al. 2021*]
- Transformers on ASR lattices and confusion networks
  - for machine translation [*Zhang et al., 2019;*]
  - and language understanding [*Liu et al., 2020*]

  focus on 'embedding' the ASR lattice or confusion network.

This work: LSTM vs Transformer LMs on ASR confusion networks.

Training LSTM LMs on Confusion Nets

Given LSTM with L layers and weights $\Theta = \{\theta^l_{\text{in}}, \theta^l_{\text{hid}}, \theta_{\text{out}}\}$

Sampling based training:
Sample one arc at a time from each confusion bin: $\bar{w} \sim p(w_t|S)$.
Cross entropy loss for training with sampled path:

$$\widehat{\Theta} = \arg\min_{\Theta} \sum_t -\log q(w_{t+1} = \bar{w}|h^L_t). \quad (1)$$

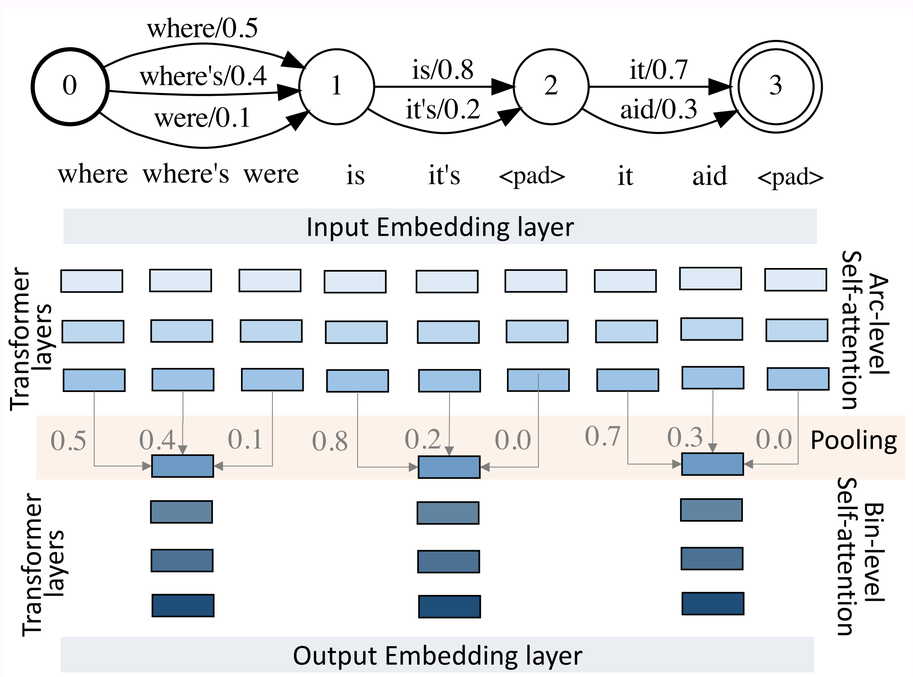
KL divergence based training:
Compute hidden state $h^1_{t,i}$ for all arcs i in a confusion bin and pool:

$$h^1_{t,i} = \sigma(\theta^1_{\text{hid}}\, h^1_{t-1} + \theta^1_{\text{in}}\, x^1_{t,i}) \quad (2)$$
$$h^1_t = \text{pool}_i(h^1_{t,i}).$$

Handle multiple outputs and uncertainty using KL loss:

$$\widehat{\Theta} = \arg\min_{\Theta} \sum_t D_{\text{KL}}\left(p(w_{t+1}|S) \,||\, q(w_{t+1}|h^L_t)\right) \quad (3)$$
$$= \arg\min_{\Theta} \sum_t \sum_{v^j} p(w_{t+1} = v^j|S) \log \frac{p(w_{t+1} = v^j|S)}{q(w_{t+1} = v^j|h^L_t)}.$$

Hierarchical Scheme for Transformer LMs



- Collapse the confusion bins into a sequence.
- Separate arc-level and bin-level layers.
- Self-attention mask based on confusion bin posteriors

$$M_{t,t'} = \begin{cases} \log p(w_{t'}) & t' \le t \\ -\infty & \text{otherwise.} \end{cases} \quad (4)$$

Experimental Setup

Dataset split	Verbmobil (VM) English		AMI *scenario* meetings	
	hours	words	hours	words
Training labeled	5.23	18 k	9.48	90 k
Training unlabeled	19.36	80 k	37.24	387 k
Development	2.14	7 k	9.77	100 k
Test	3.88	15 k	10.34	105 k

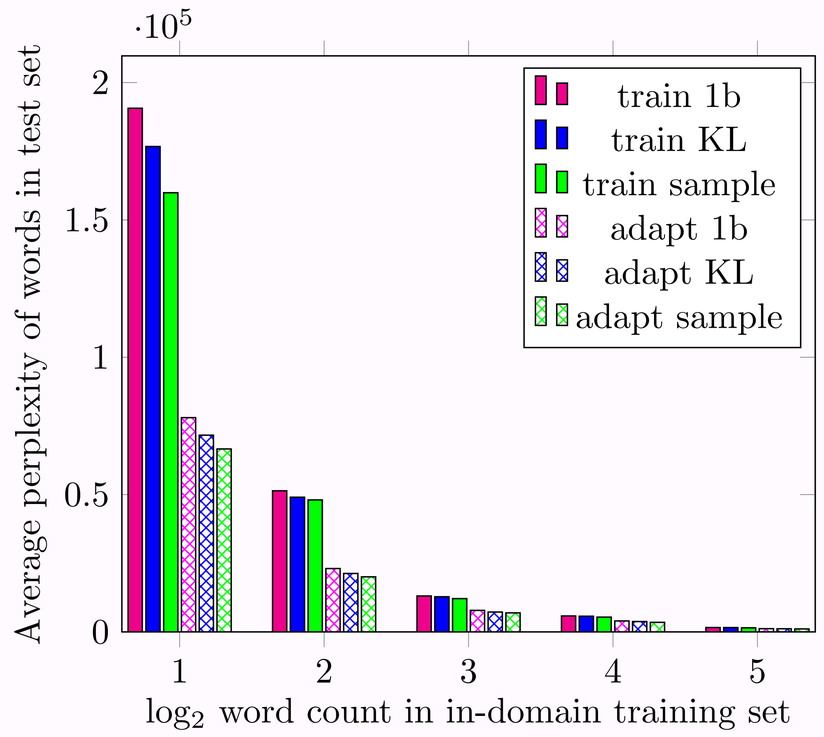
- Adaptation setting uses Switchboard text to pre-train LMs
- VM setup uses AM+LM trained on VM labeled training set
- AMI setup uses pre-trained ASpIRE ASR models
- Hyper-parameter search for Transformer LMs
- LSTM LMs have a similar number of parameters

Perplexity Evaluation

Perplexity obtained on VM and AMI test sets

LM data setup	In-domain only		Adaptation setting	
	VM	AMI	VM	AMI
LSTM LM				
lab-ref + unlab-1b	62.1	81.4	**43.1**	65.0
lab-ref + unlab-cn KL	58.9	83.2	44.3	65.5
lab-ref + unlab-cn sample	**54.7**	78.8	**43.6**	64.6
lab-ref + unlab-ref	50.4	67.9	35.4	55.2
Transformer LM				
lab-ref + unlab-1b	59.7	76.8	**43.2**	63.8
lab-ref + unlab-cn KL	64.8	78.1	44.3	64.3
lab-ref + unlab-cn KL hier.	59.6	76.2	**43.1**	**62.8**
lab-ref + unlab-cn sample	57.7	**74.2**	**43.1**	**62.5**
lab-ref + unlab-ref	47.0	63.9	38.2	53.7

Qualitative Evaluation



- Large perplexity reduction for less frequent words

Conclusion

- Lowest perplexity from sampling based training method
- Hierarchical scheme better for Transformer LMs with KL
- Transformer LMs better than LSTM LMs on AMI but not on VM
- LMs trained on confusion networks better at predicting less frequent words
- However, WER reductions are not statistically significant
- Need for more effective methods to train neural LMs on uncertain ASR hypotheses