

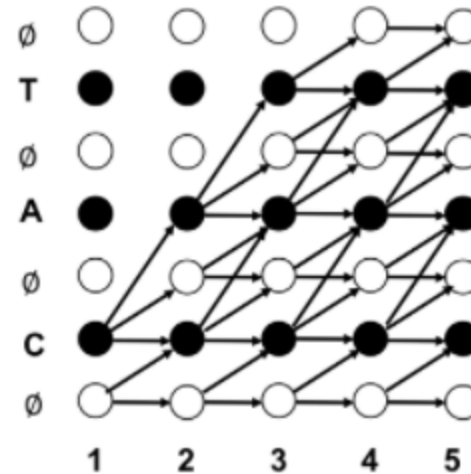
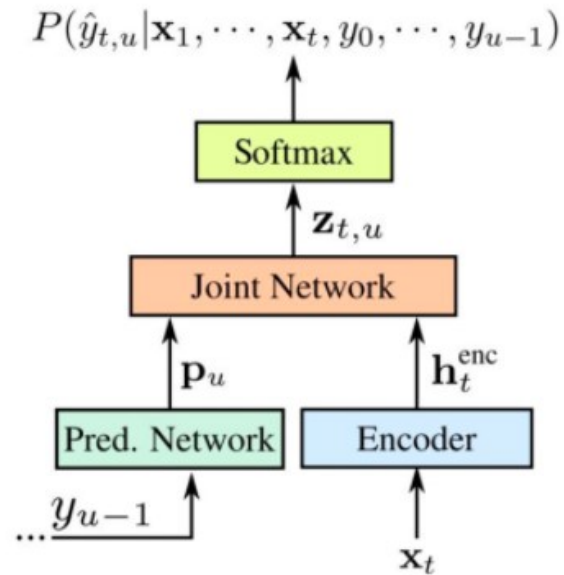
Fast Text-Only Domain Adaptation of RNN-Transducer Prediction Network

Janne Pylkkonen et al.

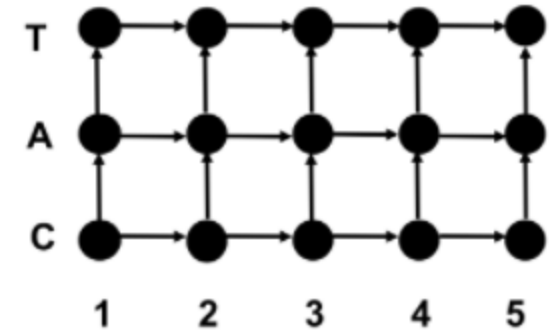
Interspeech 2021

Preliminary

- RNN-Transducer
 - Architecture (left), alignment (right)



(a) CTC



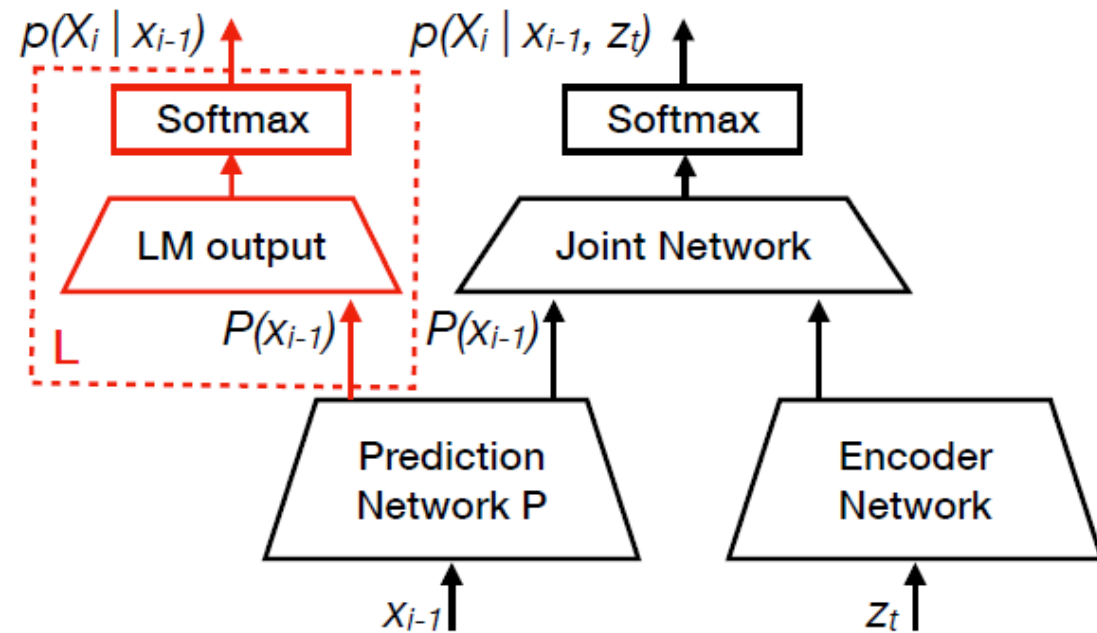
(b) RNN-Transducer

Problem

- To customize E2E models for a particular domain ..
 - Shallow fusion with external LMs
 - TTS-adaptation (fine-tune using TTS-generated data)
- However ..
 - Require changes to the model and/or decoding
 - Require generates from external data
- In this paper ..
 - They present a simple yet effective RNN-T adaptation method
 - Require only textual data

Proposed Methods

- Overview
 - (Initializing networks)
 - (RNN-T training)
- Pre-processing step
 - Attach L
 - Transcription $D_t \rightarrow p(X_i | x_{i-1})$
 - While keeping P fixed
- Adaption step
 - Transcription $D_a \rightarrow$ fine-tune
 - While keeping L fixed
 - Detach L



Proposed Methods

- Regularization
 - Only used in the adaption step ... to prevent P from over-fitting to the corpus D_a
 - Penalize changes in the predictions observed with common utterances
 - P^* : non-adapted prediction network $\rightarrow p^*(X_i|x_{i-1}) = L(P^*(x_{i-1}))$
 - D_b : set of sampled utterance \hat{x} for each adaptation example x in D_a (similar length)

$$\ell_b(x, P) = \frac{1}{n} \sum_{i=1}^n \text{KLD} (p(X_i | x_{i-1}), p^*(X_i | x_{i-1}))$$

- Penalizes the drifting of the weights of P from their original values in P^*

$$\ell_n(P) = \|P - P^*\|_2$$

- Therefore ..

$$\ell(P) = \sum_{x \in D_a} \frac{\text{CE}(x, P)}{|D_a|} + \frac{w_b}{|D_b|} \sum_{x \in D_b} \ell_b(x, P) + w_n \ell_n(P)$$

Proposed Methods

- Regularization
 - Only used in the adaption step ... to prevent P from over-fitting to the corpus D_a
 - Penalize changes in the predictions observed with common utterances
 - P^* : non-adapted prediction network $\rightarrow p^*(X_i|x_{i-1}) = L(P^*(x_{i-1}))$
 - D_b : set of sampled utterance \hat{x} for each adaptation example x in D_a (similar length)

$$\ell_b(x, P) = \frac{1}{n} \sum_{i=1}^n \text{KLD} (p(X_i | x_{i-1}), p^*(X_i | x_{i-1}))$$

- Penalizes the drifting of the weights of P from their original values in P^*

$$\ell_n(P) = \|P - P^*\|_2$$

- Therefore ..

$$\ell(P) = \sum_{x \in D_a} \frac{\text{CE}(x, P)}{|D_a|} + \frac{w_b}{|D_b|} \sum_{x \in D_b} \ell_b(x, P) + w_n \ell_n(P)$$

Setup

- ASR systems
 - Encoder: 32 dimensional filterbank energies, a convolution layer, 7 LSTM layers
 - Prediction network: 2 LSTMs
 - Joint network: a feed-forward layer, softmax activation (1000 word pieces and a blank symbol)
- + SpecAugment, beam-search decoder
- + Each encoder and prediction network is pre-initialized.
- + $W_b = 0.8$, $W_n = 0.05$
- Dataset
 - English Oscar (initialization of prediction network)
 - LibriSpeech, English Common Voice, Ted-lium 3 (RNN-T training)
 - Ted-lium 3, ATIS3, Slurp (evaluation)

Table 1: *Amount of adaptation (text) and evaluation (audio) utterances, and the size of the vocabulary in the datasets used in the experiments.*

Dataset	#Adaptation utts	#Eval utts	Vocabulary
ATIS3	6355	965	1080
Slurp	10680	4173	5168
Ted-lium	–	1155	3652

Results

Table 2: *WER for unadapted and adapted models over evaluation corpora, and relative WER reduction.*

Dataset	Unadapted	Adapted	WERR-%
ATIS3	15.9%	11.9%	-25.2%
Slurp (pooled)	42.8%	38.6%	-9.8%
Slurp (scenario)	42.8%	37.3%	-12.9%
ATIS3 (prod)	9.7%	5.4%	-44.7%
Slurp (scen; prod)	27.4%	23.4%	-14.6%

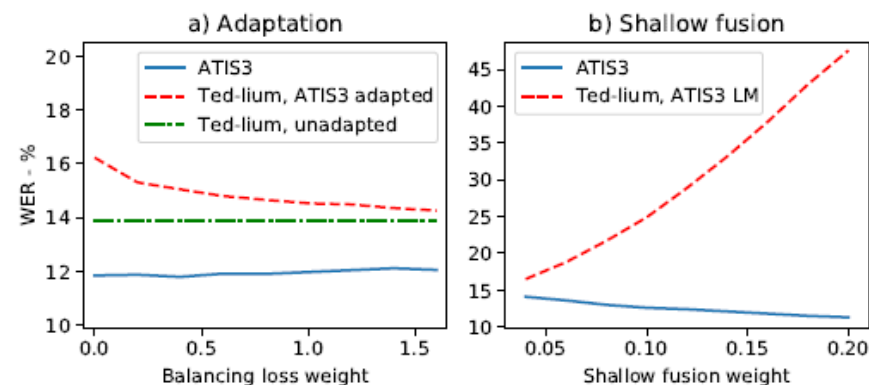


Figure 2: *Adaptation experiments with in-domain and out-of-adaptation-domain evaluation. a) The effect of the balancing loss weight to the accuracy of the adapted models. b) Shallow fusion with ATIS3 4-gram, varying the shallow fusion weight.*

Results

Table 3: *Word-level perplexities of held-out evaluation text corpora computed with the prediction network at different stages of RNN-T training and adaptation.*

Model	Perplexity		
	Oscar	LibriSp	ATIS3
#1 Initializing LM	123.6	286.7	238.4
#2 RNN-T, old LM output	151.0	292.8	276.3
#3 RNN-T, new LM output	179.8	231.4	261.9
#4 ATIS3 adapted RNN-T	197.9	251.9	23.4
RNN-T, uninitialized P	1279.2	400.5	1137.0
RNN-T, internal LM	770.5	1116.9	1055.4

Discussion

- To take the full advantage of the LM property of P and L, a new LM output layer needs to be trained after the RNN-T training.
- The initialization of the prediction network can have an important role.
- The presented RNN-T adaptation provides similar accuracy gains, but outperforms shallow fusion in the generalization capability.