Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 393–399
Marseille, 20-25 June 2022

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

Imran Sheikh[1*], Emmanuel Vincent[2], Irina Illina[2]
[1]Vivoka, 57070 Metz, France
[2]Université de Lorraine, CNRS, Inria, Loria F-54000 Nancy, France
imran.sheikh@vivoka.com, emmanuel.vincent@inria.fr, irina.illina@loria.fr

Abstract
In several ASR use cases, training and adaptation of domain-specific LMs can only rely on a small amount of manually verified text transcriptions and sometimes a limited amount of in-domain speech. Training of LSTM LMs in such limited data scenarios can benefit from alternate uncertain ASR hypotheses, as observed in our recent work. In this paper, we propose a method to train Transformer LMs on ASR confusion networks. We evaluate whether these self-attention based LMs are better at exploiting alternate ASR hypotheses as compared to LSTM LMs. Evaluation results show that Transformer LMs achieve 3–6% relative reduction in perplexity on the AMI scenario meetings but perform similar to LSTM LMs on the smaller Verbmobil conversational corpus. Evaluation on ASR N-best rescoring shows that LSTM and Transformer LMs trained on ASR confusion networks do not bring significant WER reductions. However, a qualitative analysis reveals that they are better at predicting less frequent words.

Keywords: Transformer, LSTM, language model, confusion networks, limited data

1. Introduction

Training and adaptation of domain-specific language models (LM) for automatic speech recognition (ASR) requires manually verified transcriptions of hundreds of hours of in-domain speech, and sometimes additional text from other domains. Such manually verified text resources are scarce or unavailable for most applications. In several use cases, the amount of in-domain speech data itself is limited, e.g., in the early development stages of a new application, in privacy-critical applications, or for under-resourced languages. Fully exploiting the available in-domain resources is essential in such scenarios. This motivates us to study training of LMs on a limited amount (25–50 hours) of in-domain speech data.

Early works have explored training of n-gram LMs on ASR N-best lists and lattices (Bacchiani et al., 2006; Kuznetsov et al., 2016; Levit et al., 2018) and also exploited ASR confidence scores. However, training of neural LMs on ASR hypotheses has not received attention, except in test time adaptation and conditioning of recurrent neural network (RNN) LMs (Deena et al., 2016; Gangireddy et al., 2016; Li et al., 2018). Our recent work (Sheikh et al., 2021) explored training and adaptation of Long Short Term Memory (LSTM) RNN LMs on ASR confusion networks, with the motivation of exploiting alternate uncertain ASR hypotheses obtained from limited amounts of in-domain speech. We proposed three methods, based on (1) a Kullback–Leibler (KL) divergence loss, (2) a hidden Markov model (HMM) formulation, and (3) sampling paths from the confusion networks. The sampling based method, and in some cases the KL divergence method, resulted in significant perplexity reductions as compared to training on ASR 1-best transcripts. In this paper, we extend these methods to Transformer LMs.

Transformer LMs have outperformed LSTM LMs on several large or medium-scale ASR benchmarks (Irie, 2020). The self-attention modules at different layers of the Transformer LMs have been shown to capture both local n-gram-like context as well as global information and instance specific patterns (Irie et al., 2019). We are interested in evaluating whether the self-attention mechanism of Transformers can exploit alternate hypotheses represented by ASR confusion networks, and outperform LSTM LMs in limited data setups. Prior works have extended Transformers to ASR lattices (Zhang et al., 2019; Xiao et al., 2019; Mitrofanov et al., 2021) and confusion networks (Huang and Chen, 2019; Liu et al., 2020) for machine translation (MT), ASR rescoring and spoken language understanding (SLU) tasks. In these works, a Transformer encoder embeds the lattice or confusion network into vector representations, which are then used for classification, rescoring or to generate the translated text. In contrast to these tasks, training Transformer LMs on ASR decoded graphs is more challenging since not only the input but also the output target at each step of a word sequence is not a unique class or word but a set of uncertain word hypotheses.

We propose KL divergence and sampling based methods to train Transformer LMs on ASR confusion networks. The Transformer LMs are trained in limited data setups, wherein a small amount of manual transcriptions and a limited amount of in-domain speech are available for training. We also evaluate a model adaptation setting wherein the LM is pre-trained on an out-of-domain corpus. Moreover, the performance of the Transformer LMs is compared with that of LSTM LMs that are similar in size. We would like to high-

[*]work done during postdoctoral research at Inria

light that our training methods do not modify the Transformer or LSTM architecture. The trained LSTM and Transformer LMs can be readily applied for inference or rescoring, like any other neural LM. The rest of the paper is organized as follows. Section 2 briefly recalls training of LSTM LMs on ASR confusion networks. Section 3 describes the proposed extension to Transformer LMs. Experiments and results are discussed in Section 4, followed by conclusion in Section 5.

2. Training LSTM LM on ASR confusion networks

Adopting the typical formulation of RNN LMs, for the sake of legibility, the working of LSTM LMs with L recurrent layers and weight matrices $\Theta = \{\theta^l_{\text{in}}, \theta^l_{\text{hid}}, \theta_{\text{out}}\}$ can be expressed as:

$$h^l_t = \sigma(\theta^l_{\text{hid}}\, h^l_{t-1} + \theta^l_{\text{in}}\, x^l_t) \quad (1)$$

$$q(w_{t+1}|h^L_t) = \text{softmax}(\theta_{\text{out}}\, h^L_t) \quad (2)$$

where x^1_t is the word embedding of the t-th word w_t, h^l_t is the l-th layer hidden state which encodes the history until t and $x^l_t = h^{l-1}_t$ for $l > 1$, σ is a non-linear function, and $q(w_{t+1}|h^L_t)$ is a vector of history dependent word-level LM probabilities. The LM training objective is to learn the weight matrices that minimize the cross-entropy (CE) loss:

$$\widehat{\Theta} = \arg\min_{\Theta} \sum_t -\log q(w_{t+1} = v^j|h^L_t) \quad (3)$$

where $q(w_{t+1} = v^j|h^L_t)$ is the j-th element of $q(w_{t+1}|h^L_t)$. This model assumes a single word input x^1_t at each step t in (1) and a single output target v_j at step $t+1$ in (3), hence it cannot exploit alternatives and uncertainties in ASR confusion networks. We recall two training methods from our recent work (Sheikh et al., 2021) which address this issue and result in lower perplexities than LSTM LMs trained on ASR 1-best transcripts.

2.1. KL divergence based training

To incorporate multiple confusion bin arcs at step t of the input, we modify the forward propagation in the first LSTM layer by computing individual hidden state vectors $h^1_{t,i}$ for all arcs i and pooling them as:

$$h^1_{t,i} = \sigma(\theta^1_{\text{hid}}\, h^1_{t-1} + \theta^1_{\text{in}}\, x^1_{t,i}) \quad (4)$$

$$h^1_t = \text{pool}_i(h^1_{t,i}). \quad (5)$$

The pooling can be average, weighted-sum or 1-best selection that retains the representation of the arc with highest score. The following LSTM layers are unchanged. To account for the multiple output arcs v_j at step $t+1$, we minimize the KL divergence between the LSTM LM predictions $q(w_{t+1} = v^j|h^L_t)$ and the confusion bin posteriors $p(w_{t+1} = v^j|S)$, given the speech signal S, as:

$$\begin{aligned}\widehat{\Theta} &= \arg\min_{\Theta} \sum_t D_{\text{KL}}\left(p(w_{t+1}|S) \mid\mid q(w_{t+1}|h^L_t)\right)\\ &= \arg\min_{\Theta} \sum_t \sum_{v^j} p(w_{t+1} = v^j|S)\\ &\qquad \log \frac{p(w_{t+1} = v^j|S)}{q(w_{t+1} = v^j|h^L_t)}. \end{aligned} \quad (6)$$

2.2. Sampling based training

An alternative to account for the competing hypotheses in ASR confusion networks is to sample one path at a time for each LSTM forward-backward propagation. To sample a complete path $\bar{W}$, one arc $\bar{w}_t$ can be sampled at a time based on the posterior probabilities of the arcs in each confusion bin as $\bar{w}_t \sim p(w_t|S)$. Given a sampled path from the confusion network, the LSTM LM can be trained with the standard CE loss in (3). Each training epoch sees one possible path from the ASR confusion network of each utterance. The random path for each utterance is redrawn at each epoch.

3. Training Transformer LM on ASR confusion networks

The original Transformer model (Vaswani et al., 2017) had an encoder and a decoder, each consisting of a stack of layers composed of multi-head self-attention and fully connected (FC) layers. However, the ASR LM task can be realised using either the encoder or the decoder. We adopt Transformer encoder blocks which are expected to be more powerful (Irie, 2020). The encoder blocks are similar to those of Vaswani et al. (2017). Given the input x^l_t to the l-th encoder layer, the output z^l_t of self-attention with N heads, d_x dimensional key vectors, and weights $\theta^{l,n}_Q, \theta^{l,n}_V, \theta^{l,n}_K \in \mathbb{R}^{(d_x/N)\times N}$ is obtained as

$$e^{l,n}_{t,t'} = \frac{(\theta^{l,n}_Q x^l_t)^\intercal(\theta^{l,n}_K x^l_{t'})}{\sqrt{d_x/N}} \quad (7)$$

$$\alpha^{l,n}_{t,t'} = \frac{\exp(e^{l,n}_{t,t'})}{\sum_\tau \exp(e^{l,n}_{t,\tau})} \quad (8)$$

$$z^{l,n}_t = \sum_{t'} \alpha^{l,n}_{t,t'}(\theta^{l,n}_V x^l_{t'}) \quad (9)$$

$$z^l_t = \text{Concat}(z^{l,1}_t, ..., z^{l,N}_t). \quad (10)$$

Masks are used to prevent the self-attention from using future contexts $t^{'} > t$ (Vaswani et al., 2017). The self-attention outputs go into layer normalization (LN) and FC layers along with residual connections:

$$\tilde{x}^{l+1}_t = \text{LN}(x^l_t + \text{FC}(z^l_t)) \quad (11)$$

$$x^{l+1}_t = \text{LN}(\tilde{x}^{l+1}_t + \text{FC}(\text{ReLU}(\text{FC}(\tilde{x}^{l+1}_t)))). \quad (12)$$

The outputs of the L-th layer are used to compute LM probabilities and the CE loss similar to (2) and (3), respectively. Notably, Transformer LMs can be very deep with L varying from 6 to more than 100, for datasets of different sizes (Irie, 2020).

3.1. KL divergence based hierarchical training scheme

The KL divergence based training method in Section 2.1 pools histories corresponding to multiple arcs in a confusion bin at each step t to obtain a single hidden state vector for the following step. In contrast, a Transformer LM can simultaneously attend to all the confusion-bin arcs in the history. Moreover, the self-attention can use the posterior probabilities on the arcs, as shown in previous works on MT and SLU (Zhang et al., 2019; Xiao et al., 2019; Huang and Chen, 2019; Liu et al., 2020). The approach of Liu et al. (2020) can be extended to KL divergence based training of LMs by excluding the last layer which summarizes the confusion network into one vector. This results in a rather poor performance. Hence, we explored a hierarchical scheme to train Transformer LMs on confusion networks, as illustrated in Fig. 1. The approach of Liu et al. (2020) is a special case of our hierarchical scheme, with the number of bin-level Transformer layers reduced to zero.

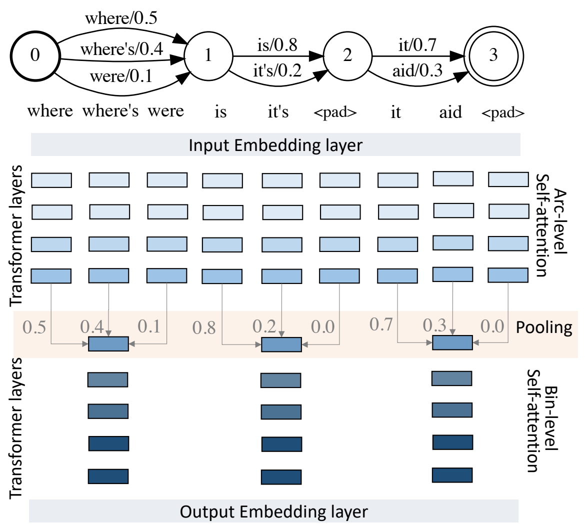

Figure 1: Proposed hierarchical scheme for training a deep Transformer LM on ASR confusion networks.

As shown in Fig. 1, the confusion network is collapsed into a long sequence which maintains the order of the confusion bins and the order of the arcs within each bin. Confusion bins with fewer arcs are padded to maintain a constant spacing. After word embedding lookup, positional embeddings are added to each arc embedding such that all arcs within a given bin have the same positional embedding. The arc embeddings pass through multiple arc-level Transformer layers. The self-attention in each arc-level Transformer layer is updated to incorporate confusion network posteriors (Zhang et al., 2019; Xiao et al., 2019; Huang and Chen, 2019; Liu et al., 2020): reusing t and t' to index the sequence of arcs obtained after collapsing the confusion network, (7) is modified as

$$e^{l,n}_{t,t'} = \frac{(\theta^{l,n}_Q x^l_t)^\intercal (\theta^{l,n}_K x^l_{t'})}{\sqrt{d_x/N}} + M_{t,t'} \tag{13}$$

$$\alpha^{l,n}_{t,t'} = \frac{\exp(e^{l,n}_{t,t'})}{\sum_\tau \exp(e^{l,n}_{t,\tau})} \tag{14}$$

$$M_{t,t'} = \begin{cases} \log p(w_{t'}) & t' \leq t \\ -\infty & \text{otherwise.} \end{cases} \tag{15}$$

Note that the arcs in a confusion bin can attend to arcs in other confusion bins from the past. The outputs of the final arc-level Transformer layer undergo pooling which fuses the outputs corresponding to all arcs in each confusion bin. The pooling can be a weighted-sum or one that retains the representation corresponding to the highest scoring arc. The outputs of the pooling operation are passed to multiple bin-level Transformer layers, that use (7). This is followed by the output embedding layer, softmax and a KL divergence loss similar to (6). It must be noted that (13)-(15) and the hierarchical scheme are only applicable for training. During decoding, there is no distinction between arc-level and bin-level layers and the forward propagation of normal text sentences through the Transformer LM follows (7)-(10).

3.2. Sampling based training

Alternatively, the sampling based training method for LSTM LMs, discussed in Section 2.2, can also be readily used to train Transformer LMs on confusion networks. Unlike the KL divergence based hierarchical training scheme, the number of computations in the sampling based training of Transformer LMs is equivalent to those in training on 1-best transcripts.

4. Experiments and Results

We evaluate Transformer LMs trained on ASR confusion networks, as discussed in Section 3, and compare them with LSTM LMs trained using the methods discussed in Section 2. As compared to our recent work (Sheikh et al., 2021), the LSTM LMs evaluated here have 1 or more LSTM layers and a parameter count that matches the Transformer LMs, as detailed in Section 4.3.

4.1. Datasets

We use two domain specific conversational speech datasets: the English subset of the Verbmobil (VM) corpus (Burger et al., 2000) and the *scenario-only* meeting subset of the AMI (Renals et al., 2007) corpus. The Verbmobil corpus contains conversations wherein the participants negotiate and agree upon an appointment schedule and/or travel plan. The AMI meeting subset contains meetings in which the participants play different roles in a design project. It must be noted that these correspond to real-world use cases for which little speech/text data is available. To simulate realistic limited data scenarios, these datasets are split into four disjoint subsets presented in Table 4.1. The labeled training set is kept small, approximately 1/4-th of the unlabeled training set. In the case of AMI, meetings ES2010, ES2016, IS1005, IS1007, TS3010, TS3011 of *SA* form our labeled training set and the remainder of *SA* forms our unlabeled training set. The development and test sets are identical to the original *scenario-only* subset. The average length of a turn in VM and AMI is 20 words and 8 words, respectively.

Split	Verbmobil (VM) English		AMI *scenario only*	
	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	7.5 k	9.77	100 k
Test	3.88	15 k	10.34	105 k

Table 1: Datasets and splits.

4.2. ASR setup

In the case of the VM dataset, a TDNN-chain acoustic model and a 3-gram LM are trained on the labeled training set (Sheikh et al., 2020). These models give a Word Error Rate (WER) of 39.52% and 39.77% on the VM development and test sets, respectively. For experiments on the AMI dataset we use the ASpIRE chain model with the already compiled HCLG (Trmal, 2019). The motivation behind this choice is to evaluate the performance with larger out-of-domain pre-trained models, in contrast to the VM setup. The ASpIRE models result in 33.15% and 35.82% WER on the AMI development and test sets, respectively.

4.3. LM training setup

LSTM and Transformer LMs are trained on the combination of the small labeled training set (**lab**) and the larger unlabeled training set (**unlab**) of VM or AMI. Accordingly, training uses manual transcriptions (**ref**) of the labeled training set and ASR hypotheses of the unlabeled training set, which can be 1-best transcriptions (**1b**) or confusion networks (**cn**). KL divergence based training of Transformer LMs using the approach of Liu et al. (2020) (**KL**) is evaluated apart from the proposed hierarchical training scheme (**KL hier.**). We evaluate training only on the limited in-domain data (VM or AMI) as well as an adaptation setting. Adaptation involves training the LSTM/Transformer LM on a combination of out-of-domain and in-domain data, followed by a fine-tuning on the in-domain data. Switchboard corpus (Godfrey et al., 1992) transcriptions are used as the out-of-domain data.

The Transformer LMs use sine and cosine based positional encodings used in the original Transformer model (Vaswani et al., 2017). To find the best Transformer LM configurations, we performed a hyper-parameter search for the number of layers and attention heads, the dimension of a layer, and the dropout and learning rates. Transformers trained only on in-domain data have about 2 M parameters with 8 layers and 8 attention heads, and those in the adaptation setting have 8 M parameters with 12 layers and 8 attention heads. The size of the Key/Query/Value vectors and FC layers were 192 for training only on in-domain data and 256 in the adaptation setting. We observed better performance with SGD than the ADAM optimizer. The SGD optimizer had an initial learning rate of 0.1, momentum 0.9 and a weight decay 1e-5. The initial learning rate for LM adaptation was lowered to 0.001.

Transformer LMs trained using KL divergence performed better when pooling was done in the pre-final layers. Pooling that retains the best arc representation turned out to be better for VM and weighted-sum pooling was better for AMI. The best performing LSTM LMs with number of parameters matching the Transformer LMs are chosen by varying the number of layers, dimensions and dropout. LSTMs in training and adaptation settings end up with 1 and 2 layers, respectively. All LMs use a tied input-output embedding matrix (Press and Wolf, 2017).

4.4. Perplexity Results

Table 2 presents perplexities obtained by the LSTM and Transformer LMs trained only on the in-domain data (VM or AMI). We use the Wilcoxon signed-rank test to ensure that the differences in perplexity are statistically significant, at $p = 0.05$ (Dror et al., 2018). Among LMs trained on ASR hypotheses, the sampling based method achieves the lowest perplexity for both LSTM and Transformer LMs. The reduction in perplexities is statistically significant as compared to training on ASR 1-best transcripts. When comparing LSTM versus Transformer LMs, we can observe that the Transformer LM achieves lower perplexities in the case of AMI but not in the case of VM. Among Transformer LMs trained using the KL divergence method, the proposed hierarchical training scheme results in lower perplexities as compared to a simple extension of the approach of Liu et al. (2020) with KL divergence loss. However, perplexity reductions from hierarchical training are not statistically significant as compared to training on ASR 1-best transcripts.

Table 3 presents perplexities obtained by LSTM and Transformer LMs in the adaptation setting. In this setting, the sampling based method leads to the lowest perplexity on the AMI dataset and it is on par with training on 1-best transcripts on the VM dataset, both for LSTM and Transformer LMs. Overall, Transformer LMs perform better than LSTM LMs on the AMI dataset but not on the VM dataset, similar to the results in Table 2. Similarly, KL divergence based training of Transformer LMs results in lower perplexities with the proposed hierarchical training scheme but it fails to outperform Transformer LMs trained on the ASR 1-best transcripts.

LM setup	VM dev	VM test	AMI dev	AMI test
LSTM LM				
lab-ref + unlab-1b	58.8	62.1	72.8	81.4
lab-ref + unlab-cn KL	55.2	58.9	73.6	83.2
lab-ref + unlab-cn sample	**52.3**	**54.7**	71.1	78.8
lab-ref + unlab-ref	47.6	50.4	61.3	67.9
Transformer LM				
lab-ref + unlab-1b	56.7	59.7	68.6	76.8
lab-ref + unlab-cn KL	61.7	64.8	70.0	78.1
lab-ref + unlab-cn KL hier.	56.2	59.6	68.4	76.2
lab-ref + unlab-cn sample	54.6	57.7	**66.5**	**74.2**
lab-ref + unlab-ref	45.0	47.0	57.3	63.9

Table 2: Perplexity of LSTM and Transformer LMs trained only on the in-domain data. Bold font indicates lowest perplexity and performance statistically similar to it.

LM setup	VM dev	VM test	AMI dev	AMI test
LSTM LM				
lab-ref + unlab-1b (pre)	63.4	63.2	90.7	97.3
lab-ref + unlab-1b	**40.9**	**43.1**	59.5	65.0
lab-ref + unlab-cn KL	42.2	44.3	60.3	65.5
lab-ref + unlab-cn sample	**41.3**	**43.6**	58.8	64.6
lab-ref + unlab-ref (pre)	52.6	53.4	82.6	88.0
lab-ref + unlab-ref	34.0	35.4	50.8	55.2
Transformer LM				
lab-ref + unlab-1b (pre)	47.8	48.3	67.5	73.1
lab-ref + unlab-1b	**41.8**	**43.2**	**57.4**	63.8
lab-ref + unlab-cn KL	43.1	44.3	58.2	64.3
lab-ref + unlab-cn KL hier.	**41.6**	**43.1**	**57.2**	**62.8**
lab-ref + unlab-cn sample	**41.8**	**43.1**	**56.7**	**62.5**
lab-ref + unlab-ref (pre)	44.5	45.1	58.1	62.3
lab-ref + unlab-ref	37.3	38.2	48.8	53.7

Table 3: Perplexity of LSTM and Transformer LMs in the adaptation setting. The models indicated as 'pre' are those which have not been fine-tuned. Bold font highlights lowest perplexity and performance statistically similar to it.

4.5. Qualitative Analysis

We conducted a qualitative analysis to further understand the reduction in perplexities obtained by the LMs trained on the ASR confusion networks. The analysis compares the perplexities obtained by the LMs, trained with different methods, for the less frequent words. In this analysis, we computed the perplexity assigned by an LM to the words in the test set, then grouped the words based on their count in the in-domain training set and plotted the average perplexity obtained by each group of words.

Figure 2 shows the average perplexity obtained by the Transformer LMs on the AMI test set words, wherein the X-axis represents word groups based on their counts in the AMI in-domain training set. Transformer LMs trained only on the in-domain data are denoted by prefix 'train' and Transformer LMs in the adaptation setting are denoted by prefix 'adapt'. We can observe that KL and sampling based training methods result in lower perplexity for less frequent words, in the case of training only on the in-domain data. Moreover, the sampling-based method results in a lower perplexity than KL based training for these less frequent words. In the adaptation setting, the reduction in perplexity obtained from the KL and sampling based methods is smaller. This is mainly because Transformer LMs from the adaptation setting have already seen most less frequent words during the pre-training on out-of-domain data. Overall, we can conclude that the proposed methods to train neural LMs on ASR confusion networks can be better at predicting less frequent words.

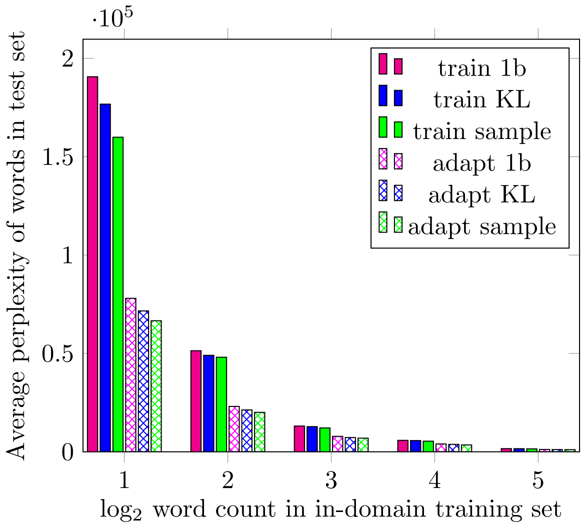

Figure 2: Average perplexity of Transformer LMs on words in the AMI test set. (Higher count words on the X-axis are not shown.)

4.6. WER Results

Following the perplexity evaluation, we present a WER evaluation of the different LSTM and Transformer LMs. In order to perform this WER evaluation, lattices obtained from the first pass decoding were rescored using a 3-gram LM trained on the LM data setup: lab-ref + unlab-1b. 100-best lists were obtained from these lattices and rescored using the LSTM or Transformer LMs. We present WER results for the setting wherein the LMs are trained only on the in-domain data, as we observe significant reductions in perplexity in this case (see Table 2). The WER results obtained for this setting are shown in Table 4. The matched pairs sentence-segment word error test (Gillick and Cox, 1989) from the NIST scoring toolkit[1] is used to ensure that the differences in WER are statistically significant (at $p = 0.05$).

[1]`https://github.com/usnistgov/SCTK`

LM setup	VM dev	VM test	AMI dev	AMI test
3g decode LM				
lab-ref	39.5	39.7	32.2	35.1
LSTM LM				
lab-ref + unlab-1b	36.2	**36.3**	**31.1**	**33.7**
lab-ref + unlab-cn KL	**35.7**	**36.1**	**31.3**	**33.8**
lab-ref + unlab-cn sample	**35.9**	**36.1**	**31.2**	**33.7**
lab-ref + unlab-ref	32.9	32.9	29.3	31.9
Transformer LM				
lab-ref + unlab-1b	**36.2**	36.7	**30.9**	**33.5**
lab-ref + unlab-cn KL hier.	38.0	38.3	34.4	37.2
lab-ref + unlab-cn sample	**36.2**	**36.4**	**31.1**	**33.6**
lab-ref + unlab-ref	32.6	32.7	29.5	32.0

Table 4: WER of LSTM and Transformer LMs trained only on the in-domain data. Bold font indicates lowest WER and performance statistically similar to it.

We find that the sampling and KL based training methods do not achieve any significant WER reductions as compared to the training on 1-best hypotheses, for both LSTM and Transformer LMs. This could be due to the fact that the perplexity reductions come mainly from the less frequent words which do not contribute significantly in the ASR evaluation. Finally we also note that only part of WER gap between the baseline and the topline LMs has been filled by the LMs trained on the ASR hypotheses of the unlabelled data. This motivates the need for more effective methods to train LSTM and Transformer LMs on uncertain ASR hypotheses.

5. Conclusion

We presented methods to train Transformer LMs on ASR confusion networks, in scenarios having a limited amount of in-domain speech. KL divergence based training with a hierarchy of arc-level and bin-level layers results in significant reduction in perplexities, as compared to training with only arc-level layers. However, the resulting Transformer LMs are on par with those trained on ASR 1-best transcripts. The sampling based training method results in the lowest perplexities for both Transformer and LSTM LMs. Overall, Transformer LMs gave lower perplexity than LSTM LMs on the AMI scenario meetings but not on the VM conversations. N-best rescoring with the LSTM and Transformer LMs shows that the LMs trained on ASR confusion networks do not achieve significant reductions in WER as compared to those trained on the 1-best hypothesis. However, a qualitative analysis reveals that the LMs trained on ASR confusion networks using KL divergence and sampling based methods can be better at predicting less frequent words.

6. Acknowledgments

This work was supported by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 825081 COMPRISE. Experiments were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`).

7. Bibliographical References

Bacchiani, M., Riley, M., Roark, B., and Sproat, R. (2006). MAP adaptation of stochastic grammars. *Computer Speech and Language*, 20(1):41–68.

Burger, S., Weilhammer, K., Schiel, F., and Tillmann, H. G. (2000). Verbmobil data collection and annotation. In *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 537–549.

Deena, S., Hasan, M., Doulaty, M., Saz, O., and Hain, T. (2016). Combining feature and model-based adaptation of RNNLMs for multi-genre broadcast speech recognition. In *Proceedings of Interspeech*, pages 2343–2347.

Dror, R., Baumer, G., Shlomov, S., and Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, July.

Gangireddy, S. R., Swietojanski, P., Bell, P., and Renals, S. (2016). Unsupervised adaptation of recurrent neural network language models. In *Proceedings of Interspeech*, pages 2333–2337.

Gillick, L. and Cox, S. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 532–535.

Godfrey, J. J., Holliman, E. C., and McDaniel, J. (1992). Switchboard: telephone speech corpus for research and development. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 517–520.

Huang, C.-W. and Chen, Y.-N. (2019). Adapting pretrained transformer to lattices for spoken language understanding. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 845–852.

Irie, K., Zeyer, A., Schlüter, R., and Ney, H. (2019). Language modeling with deep transformers. In *Proceedings of Interspeech*, pages 3905—3909.

Irie, K. (2020). *Advancing neural language modeling in automatic speech recognition*. Ph.D. thesis, RWTH Aachen University.

Kuznetsov, V., Liao, H., Mohri, M., Riley, M., and Roark, B. (2016). Learning n-gram language models from uncertain data. In *Proceedings of Interspeech*, pages 2323–2327.

Levit, M., Parthasarathy, S., and Chang, S. (2018). What to expect from expected Kneser-Ney smoothing. In *Proceedings of Interspeech*, pages 3378–3382.

Li, K., Xu, H., Wang, Y., Povey, D., and Khudanpur, S. (2018). Recurrent neural network language model adaptation for conversational speech recognition. In *Proceedings of Interspeech*, pages 3373–3377.

Liu, C., Zhu, S., Zhao, Z., Cao, R., Chen, L., and Yu, K. (2020). Jointly encoding word confusion network and dialogue context with BERT for spoken language understanding. In *Proceedings of Interspeech 2020*, pages 871–875.

Mitrofanov, A., Korenevskaya, M., Podluzhny, I., Khokhlov, Y., Laptev, A., Andrusenko, A., Ilin, A., Korenevsky, M., Medennikov, I., and Romanenko, A. (2021). LT-LM: A novel non-autoregressive language model for single-shot lattice rescoring. In *Proceedings of Interspeech*, pages 4039–4043.

Press, O. and Wolf, L. (2017). Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL): Volume 2, Short Papers*, pages 157–163.

Renals, S., Hain, T., and Bourlard, H. (2007). Recognition and understanding of meetings the AMI and AMIDA projects. In *Proceedings of IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, pages 238–247.

Sheikh, I., Vincent, E., and Illina, I. (2020). On semi-supervised LF-MMI training of acoustic models with limited data. In *Proceedings of Interspeech*, pages 986–990.

Sheikh, I., Vincent, E., and Illina, I. (2021). Training RNN language models on uncertain ASR hypotheses in limited data scenarios. submitted to *Computer Speech and Language*, available at `https://hal.inria.fr/hal-03327306`, August.

Trmal, Y. (2019). ASpIRE chain model. Online: `http://kaldi-asr.org/models/m1`, Last Accessed: September 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 6000–6010.

Xiao, F., Li, J., Zhao, H., Wang, R., and Chen, K. (2019). Lattice-based transformer encoder for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3090–3097, July.

Zhang, P., Ge, N., Chen, B., and Fan, K. (2019). Lattice transformer for speech translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6475–6484, July.