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

Imran Ahamad Sheikh, Emmanuel Vincent, Irina Illina

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.

Index Terms— Transformer, language model, confusion networks

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 [1, 2, 3]. 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 [4, 5, 6]. Our recent work [7] 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 [8]. 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 [9]. 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 [10, 11, 12] and confusion networks [13, 14] 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. The rest of the paper is organized as follows. Section 2 quickly recalls how to train LSTM LMs on 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.

[*]work done during postdoctoral research with Inria Nancy

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 [7] 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 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 following 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 $p(w_{t+1} = v^j|S)$ as:

$$\widehat{\Theta} = \arg\min_{\Theta} \sum_t D_{\text{KL}}\left(p(w_{t+1}|S)\ ||\ q(w_{t+1}|h^L_t)\right) \quad (6)$$

$$= \arg\min_{\Theta} \sum_t \sum_{v^j} p(w_{t+1} = v^j|S) \log \frac{p(w_{t+1} = v^j|S)}{q(w_{t+1} = v^j|h^L_t)}$$

where S denotes the observed speech signal.

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 [15] 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 [8]. The encoder blocks are similar to those in [15]. Given the input x^l_t to the l-th encoder layer, the output z^l_t of self-attention with N heads 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

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

$$z^{l,n}_t = \sum_{t'} \alpha^{l,n}_{t,t'} (\theta^{l,n}_V x^l_{t'}); \quad z^l_t = \text{Concat}(z^{l,1}_t, ..., z^{l,N}_t). \quad (8)$$

Masks are used to prevent the self-attention from using future contexts $t' > t$ [15]. The self-attention outputs go into layer normalization and FC layers along with residual connections:

$$\tilde{x}^{l+1}_t = \text{LayerNorm}(x^l_t + \text{FC}(z^l_t)) \quad (9)$$

$$x^{l+1}_t = \text{LayerNorm}(\tilde{x}^{l+1}_t + \text{FC}(\text{ReLU}(\text{FC}(\tilde{x}^{l+1}_t)))). \quad (10)$$

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 [8].

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 [10, 11, 13, 14]. The approach in [14] 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 in [14] is a special case of our hierarchical scheme, with the number of bin-level Transformer layers reduced to zero.

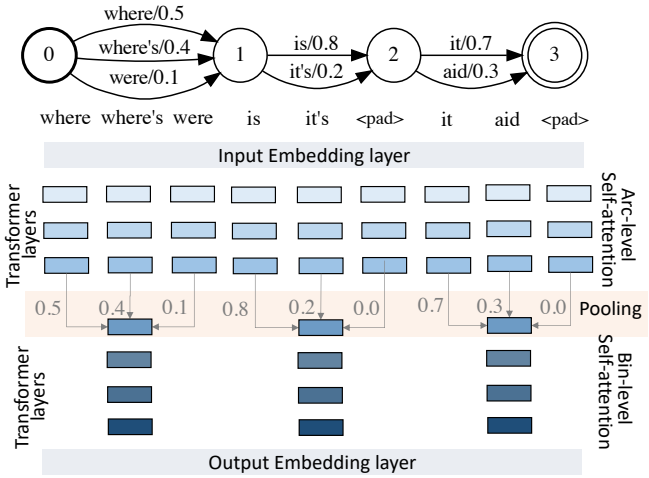

Fig. 1. Proposed hierarchical scheme for training a deep Transformer LM on ASR confusion networks.

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 position 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 [10, 11, 13, 14]: reusing t and $t^{'}$ to index the sequence of arcs obtained after collapsing the confusion network, (7) is modified as

$$e^{l,n}_{t,t'} = \frac{(\theta^{l,n}_{Q} x^{l}_{t})^{\intercal}(\theta^{l,n}_{K} x^{l}_{t'})}{\sqrt{d_x/N}} + M_{t,t'} \; ; \; \alpha^{l,n}_{t,t'} = \frac{\exp(e^{l,n}_{t,t'})}{\sum_{\tau}\exp(e^{l,n}_{t,\tau})}$$

$$M_{t,t'} = \begin{cases} \log p(w_{t'}) & t^{'} \leq t \\ -\infty & \text{otherwise.} \end{cases} \quad (11)$$

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 (11) and the hierarchical scheme are only applicable for training.

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 [7], 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 [16] and the *scenario-only* meeting subset of the AMI [17] corpus. 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 [18]. 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 [19]. 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 in [14] (**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 [20] transcriptions are used as the out-of-domain data.

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 and 8 layers, and those in the adaptation setting have 8 M parameters and 12 layers. 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 [21].

4.4. Performance evaluation

Table 2 presents perplexities obtained by the LSTM and Transformer LMs trained only on the in-domain data (VM or AMI). 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 [14] with KL divergence loss. However, perplexity reductions from hierarchical training are not statistically significant as compared to training on ASR 1-best transcripts.

	LM setup	VM dev	VM test	AMI dev	AMI test
LSTM	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	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.

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	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	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.

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 performed better than LSTM LMs on the AMI scenario meetings but not on the VM conversations.

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.

7. REFERENCES

[1] Michiel Bacchiani, Michael Riley, Brian Roark, and Richard Sproat, "MAP adaptation of stochastic grammars," *Computer Speech and Language*, vol. 20, no. 1, pp. 41–68, 2006.

[2] Vitaly Kuznetsov, Hank Liao, Mehryar Mohri, Michael Riley, and Brian Roark, "Learning n-gram language models from uncertain data," in *Proceedings of Interspeech*, 2016, pp. 2323–2327.

[3] Michael Levit, Sarangarajan Parthasarathy, and Shuangyu Chang, "What to expect from expected Kneser-Ney smoothing," in *Proceedings of Interspeech*, 2018, pp. 3378–3382.

[4] Salil Deena, Madina Hasan, Mortaza Doulaty, Oscar Saz, and Thomas Hain, "Combining feature and model-based adaptation of RNNLMs for multi-genre broadcast speech recognition," in *Proceedings of Interspeech*, 2016, pp. 2343–2347.

[5] Siva Reddy Gangireddy, Pawel Swietojanski, Peter Bell, and Steve Renals, "Unsupervised adaptation of recurrent neural network language models," in *Proceedings of Interspeech*, 2016, pp. 2333–2337.

[6] Ke Li, Hainan Xu, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Recurrent neural network language model adaptation for conversational speech recognition," in *Proceedings of Interspeech*, 2018, pp. 3373–3377.

[7] Imran Sheikh, Emmanuel Vincent, and Irina Illina, "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`, Aug. 2021.

[8] Kazuki Irie, *Advancing neural language modeling in automatic speech recognition*, Ph.D. thesis, RWTH Aachen University, 2020.

[9] Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney, "Language modeling with deep transformers," in *Proceedings of Interspeech*, 2019, pp. 3905—3909.

[10] Pei Zhang, Niyu Ge, Boxing Chen, and Kai Fan, "Lattice transformer for speech translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, July 2019, pp. 6475–6484.

[11] Fengshun Xiao, Jiangtong Li, Hai Zhao, Rui Wang, and Kehai Chen, "Lattice-based transformer encoder for neural machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, July 2019, pp. 3090–3097.

[12] Anton Mitrofanov, Mariya Korenevskaya, Ivan Podluzhny, Yuri Khokhlov, Aleksandr Laptev, Andrei Andrusenko, Aleksei Ilin, Maxim Korenevsky, Ivan Medennikov, and Aleksei Romanenko, "LT-LM: A novel non-autoregressive language model for single-shot lattice rescoring," in *Proceedings of Interspeech*, 2021, pp. 4039–4043.

[13] Chao-Wei Huang and Yun-Nung Chen, "Adapting pretrained transformer to lattices for spoken language understanding," in *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 845–852.

[14] Chen Liu, Su Zhu, Zijian Zhao, Ruisheng Cao, Lu Chen, and Kai Yu, "Jointly encoding word confusion network and dialogue context with BERT for spoken language understanding," in *Proceedings of Interspeech 2020*, 2020, pp. 871–875.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 6000–6010.

[16] Susanne Burger, Karl Weilhammer, Florian Schiel, and Hans G. Tillmann, "Verbmobil data collection and annotation," in *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 537–549. 2000.

[17] Steve Renals, Thomas Hain, and Herve Bourlard, "Recognition and understanding of meetings the AMI and AMIDA projects," in *Proceedings of IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, 2007, pp. 238–247.

[18] Imran Sheikh, Emmanuel Vincent, and Irina Illina, "On semi-supervised LF-MMI training of acoustic models with limited data," in *Proceedings of Interspeech*, 2020, pp. 986–990.

[19] Yenda Trmal, "ASpIRE chain model," Online: `http://kaldi-asr.org/models/m1`, Last Accessed: September 2021.

[20] John J. Godfrey, Edward C. Holliman, and Jane McDaniel, "Switchboard: telephone speech corpus for research and development," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1992, pp. 517–520.

[21] Ofir Press and Lior Wolf, "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*, Apr. 2017, pp. 157–163.