Investigating the scarce data and resources problem for speech recognition using transfer learning and data augmentation

Fahrettin Gökgöz[1], *Mahmoud Hashem*[1]

[1]*Fraunhofer FKIE*

@fkie.fraunhofer.de

Abstract: We investigate the effect of the transfer learning procedure on e2e Automatic Speech Recognition systems using a limited amount of data. We use a DeepSpeech inspired base-line in our experiments and based on different transfer learning techniques. Our experimental results indicate the benefit of the augmented progressive transfer method in minimizing the over-fitting and improving the accuracy.

1 Introduction

The ever-growing number of voice assistants, auto-captioning, and voice-search tools relies on one critical component: automatic speech recognition (ASR). Classical ASR systems consist of complex, heavily engineered approaches and require specialized input features and acoustic models [1]. Improvement of such a pipeline requires domain experts' considerable amount of time. In recent years various architectural enhancements in deep neural networks contributed to substantial progress within many research fields including ASR [2].

Nowadays, state of the art speech recognition performance on scientific data-sets is achieved by some end to end (e2e) models using deep learning [3], such as DeepSpeech [4, 5]. As with all deep learning techniques, these models highly depend on data availability. Unfortunately, this leads to problems in adapting e2e approaches to scarce/low-resource languages.

Transfer learning provides a promising approach in this direction [6, 7]. However, the naive way of adapting all weights of neural networks for a small amount of data allows significant changes in the learned feature characteristics and leads to over-fitting [1]. Mirsamadi and Hansen [8] propose to add a linear layer for solving this problem, whereas Sainath et al. [9] suggest adding a second decoder for the constrained-based training [10]. Though, they require some modifications to the base model.

In this paper, we propose a progressive transfer method for transfering an ASR model from a source to a target language. The method only allows layer-by-layer adjustment so the pre-trained network is only updated starting from the last layer during the transfer learning phase. The source language performance is used as a regularizer during weight modifications on the current layer. The previous layer is only allowed to be updated with a lower learning rate after reaching the mutual best results on both source and target language on the current layer. Since capturing enough characteristics from the target language is playing a critical role during transfer learning, a data augmentation method [11] is added during this phase.

We present our results from the experiments that are conducted on English and German data-sets showing the efficiency of the progressive transfer learning method. The results give insights on to what extend features learned from English data can influence German training in both time and accuracy. Section 2 summarizes the methods that are used in the design of the network and training, and Section 3 outlines our experiment design and results are shown in Section 4, and finally paper is concluded in Section 5.

2 Methods

2.1 Model

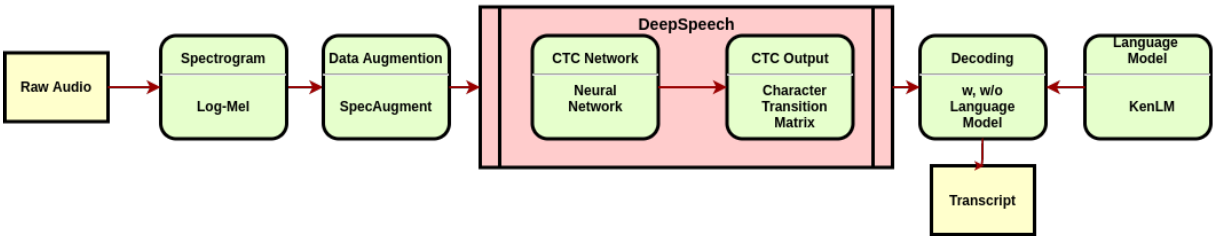

Figure 1 – Experimental End-to-End System Design with all components, First the raw audio is processed to Mel-spectrogram, Then the data Augmentation component applies some deformations. Deepspeech module processes deformed Mel-spectrogram to a character transition matrix, That finally used to decode with or without a language model the most probable transcript.

Our model, shown in Figure 1, inspired by Deepspeech [4, 5], is a recurrent neural network (RNN) that generates character sequences transcription from Mel-Spectrograms. Each training example is formulated as (x_i, y_i) that correspond to a single utterance x_i and its label y_i respectively, which is sampled from dataset $X = \{(x_1, y_1), (x_2, y_2), ...\}$. x_i, is a time-series vector of length T_i that encodes the audio features in each timestep. Mel-spectrogram's features are used as our input, so $x_i(t, p)$ denotes the p's frequency bin power in the audio frame at time t. the model then produces a sequence of character probabilities as transcription candidates for the input spectrogram, where the output at each timestep is calculated as $\hat{y}(t) = P(c_t|x), c \in Alphabet$.

The Model has a six-layer architecture, where the first three layers are feed forwards, then follows a unidirectional LSTM [12] layer followed by two additional feed-forward layers. We denote individual layer l, where $h(l)$ is the output of layer l, knowing that $h(0)$ corresponds to an input.

All of the non-recurrent layers operations are represented by equation 1, knowing that the first layer, h_t^0 corresponds to the spectrogram frame x_t.

$$h_t^{(l)} = g(W^{(l)} h_t^{(l-1)} + b^{(l)}) \tag{1}$$

The parameters $W^{(l)}$ and $b^{(l)}$ in the equation 1 are representing the weights and biases for the corresponding layer. After that, the clipped rectified-linear unit (ReLu) is used as the activation function denoted as $g(z) = min\{max\{0, z\}, 20\}$.

The fourth layer is a unidirectional LSTM layer, which is represented in equation 2, Where the LSTM function is described at [12]

$$h_t^{(4)} = g(LSTM(h_t^{(3)}, h_{(t-1)}^{(4)}, C_{(t-1)}^{(4)})) \tag{2}$$

the predicted character probabilities for each time slice t and character k in the alphabet is calculated by applying softmax to the final layer as represented in equation 3

$$\hat{y}_{t,k} \equiv \mathbb{P}(c_t = k|x) = \frac{\exp(h_t^{(6)}(k))}{\sum_{j \in J} \exp(h_t^{(6)}(j))} \tag{3}$$

Finally, we use CTC [13] to measure the loss $L(\hat{y}, y)$ between the predicted character probabilities $\hat{y}$ and the ground truth y with respect to the network outputs and backpropagate the gradient $\nabla L(\hat{y}, y)$ to all of the model parameters through the rest of the network.

2.2 Data Augmentation

Diversity of the augmentation helps the network to learn better features and also minimizes the over-fitting risk [14]. Mel-Spectrograms are used as input to our network; we apply deformations on those features to provide the capability to enrich the data during the training time. Augmentation deformations should be including segment-based time and frequency masking as presented in [15]. The following operations are applied as data augmentation during our training procedure.

1. Time warping views the Mel-Spectrogram as an image where the time axis is horizontal and the frequency axis is vertical with τ time steps, and it is applied by choosing a random point along the horizontal axis in the temporal segment $(W, \tau - W)$ and the direction of warping either to the left or to the right. Additionally, warping distance w is sampled from a uniform distribution with a mean of 0 and a standard deviation of W.

2. Frequency masking masks f consecutive Mel-frequency channels $[f_0, f_0 + f)$, where f0 is chosen from $[0, \upsilon - f)$, and f is sampled from a uniform distribution with a mean of 0 and a Standard deviation of F. υ is the number of Mel-frequency channels.

3. Time masking masks t consecutive time steps $[t_0, t_0 + t)$, where t0 is chosen from $[0, \tau - t)$, and t sampled from a uniform distribution with a mean of 0 and a standard deviation of T.

2.3 Language Model

The perceived performance of the ASR system depends on both acoustic and language models. Measuring the added value of the acoustic model in the overall accuracy is another criterion for our experiments. We employ a probabilistic language model to be able to observe this. In particular, and we selected KenLM [16, 17] because of its scalability, short query time, and memory efficiency.

3 Setup

In this paper, we try to investigate the effect of the training procedure on transfer learning. Lack of data in the target language and a large parameter space cause over-fitting and reduce performance. Therefore, our experiment setup focuses on observing the individual regularization's effect on the transfer. For this purpose, we start from a "full transfer learning" as a baseline, which allows unconstrained training for all of the layers. Then we observe the effect of the following options.

The first option in our experiments is "mean transfer learning" . In this setup, we freeze feed-forward layers before LSTM and only allow the training on the rest of the model. In this experiment, we aim to observe the effect of previously extracted features with the combination of the newly learned temporal feature relations.

The second option in our experiments is "gradual transfer learning" . It is sequentially allowing the layers to be trained and unfreezing a layer once its successor is converged. This experiment is designed to capture the information gain from the trained source language and the effect of sequentially allowing the injection of the target language data to those layers. In this case, we are keeping the learning rate stable.

The final option in our experiment is "progressive transfer learning", we follow the same procedure as in the previous one except that the learning rate is reduced progressively.

Each experiment described above looks at the problem from a training procedural regularization perspective, and the complementary method is data augmentation. Therefore all the experiments described above were also conducted with frequency masking, time warping, and time masking data augmentations combined to observe their effect on training.

3.1 Resources

3.1.1 Data

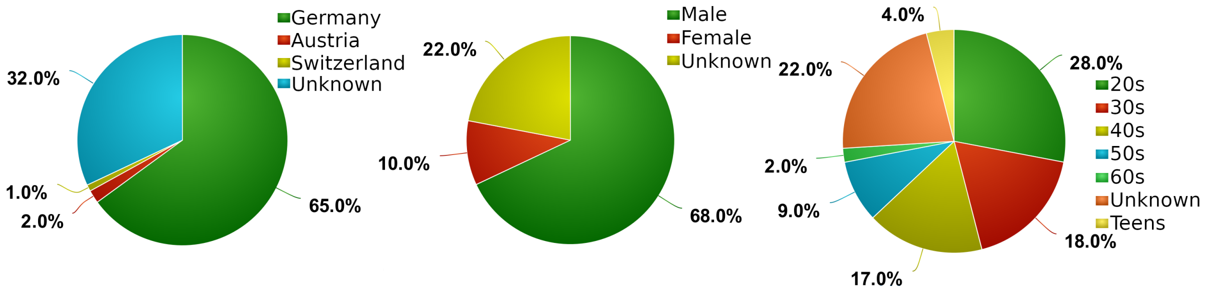

Figure 2 – CV data distribution per geographical location, gender, and age groups in respective order

The Common Voice [18] (CV) is a community effort open-source data-set that is under active development and maintenance by Mozilla. It is a multilingual collection, and its latest version contains 59 different languages, which aims to help improving speech technology relevant research and the development of the tools. Fundamentally the idea is crowd-sourcing the utterances collection and validation processes for ASR. Already having collected tens of thousands of utterances and considering the variety of languages makes it an adequate candidate for other tasks besides ASR. To the best of our knowledge, it is both the largest and most diverse audio data-set in terms of collected hours and regarding the number of speakers in the public domain for ASR tasks.

3.1.2 Baseline system

The baseline model in our experiments is inspired by DeepSpeech [4], which was described in section 2.1. It maps each 32 log mel spectrogram features to alphabet characters probabilities. All layers are consisting of 2048 hidden nodes. 40% dropout ratio is used except for the LSTM layer and the layers following it. The model was trained iteratively for a total of 325 epochs without early stopping while gradually reducing the learning rate on Fisher, LibriSpeech [19], Switchboard [20], Common Voice English, and approximately 1700 hours of transcribed WAMU (NPR) radio shows used as training corpora. During the training, the learning rate was gradually decreasing by the ratio of 0.1 after every 100 epochs. Finally, the model was optimized using ADAM Optimizer [21].

3.1.3 Hyperparameters

Our experiments can be categorized into four types namely "full transfer learning", "mean transfer learning", "gradual transfer learning" and "progressive transfer learning". Additionally, each of the experiment types is further parametrized with and without augmentation and a language model. In each of the experiments, the optimizer is Adam. Language Model weight is 0.93, and word insertion weight is 1.18 for the language model enabled experiments. Three intervals each with two frequency bandwidths masking are used for frequency masking. Three intervals each with ten milliseconds windows are used for time masking. 10% stretching/squeezing is applied to random consecutive blocks as bi-cubic time warping in data augmentation enabled experiments. The CTC's beamwidth value is 1024, the batch size is 128, and the number of training epochs is 75. An early stop was not used in full and mean transfer experiments, but learning rate decay is enabled with the sensitivity of 0.05. Ten epoch patience and 0.05 sensitivity early stop is used in gradual and progressive transfer learning experiments to unfreeze layers iteratively. Additionally, the learning rate is halved for each opened layer in the progressive transfer learning experiment.

4 Results and Discussion

	w/o A, w/o L	**w A, w/o L**	**w/o A, w L**	**w A, w L**
Full	0.28599	0.275194	0.232108	0.231583
Mean	0.289109	**0.26781**	0.259603	0.220897
Gradually	0.285466	0.270942	**0.221055**	0.214099
Progressive	**0.283566**	0.269247	0.222753	**0.207041**

Table 1 – Character Error Rates for each individual Transfer Learning Procedure

Table 1 shows character error rate (CER) for the transfer learning experiments with the learning rate 0.0001.

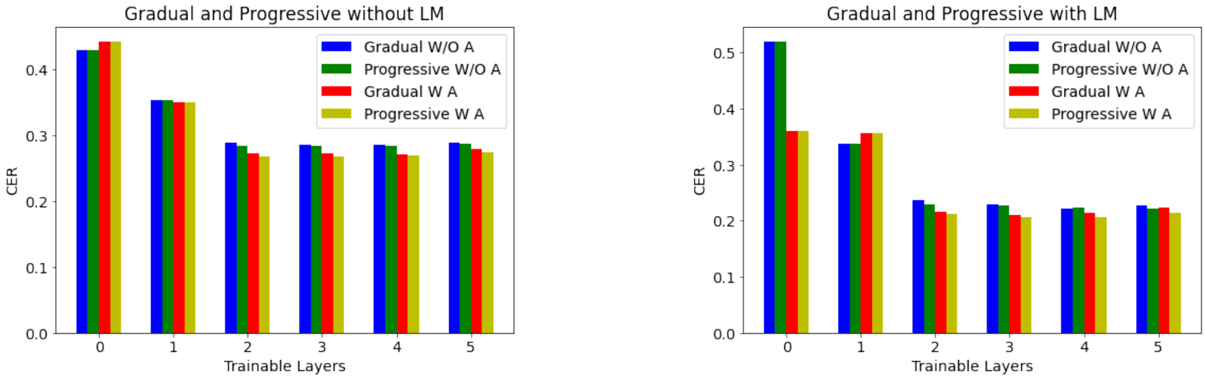

Figure 3 – Learning Rate 0.0001 - Test Results for Progressive and Gradual with/with out Language Model and Augmentation. Individual numbers on trainable layers represent the allowed layers during training in reverse order.

Figure 3 shows the results on both gradual and progressive transfer methods with the allowed trainable layers for all four combinations of using language model and augmentation. Consistently, progressive transfer learning has a positive margin over the gradual transfer learning method. Augmentation shows improvement as well in both language model enabled and disable modes. As noticed CER is comparable between mean and full transfer: this indicates that the features learned by earlier layers for the English language can be used without further change for the German language. Additional experiments were carried out with a variety of learning rates as presented by the diagrams shown in the appendix to support the consistency of the reported behaviors.

5 Conclusions

In this paper, we investigated the effect of augmented progressive transfer learning by the experimental comparison of this approach to gradually, full, and mean transfer learning methods on English to German Languages. In our experiments, we consistently achieved marginal improvement compared to other methods. Knowing that the German language does not count as a low resource language, we used a small amount of data to emulate the behavior. Yet the question of applicability on more languages, dialects, geographics are still open question for investigation, as well as the determining the right amount of data for transfer.

References

[1] Zhang, Z., J. Geiger, J. Pohjalainen, A. Mousa, and B. Schuller: *Deep Learning for Environmentally Robust Speech Recognition: An Overview of Recent Developments*. *ACM Transactions on Intelligent Systems and Technology*, 9, 2017. doi:10.1145/3178115.

[2] Malik, M., M. K. Malik, K. Mehmood, and I. Makhdoom: *Automatic speech recognition: a survey*. *Multimedia Tools and Applications*, 2020. doi:10.1007/s11042-020-10073-7. URL `http://link.springer.com/10.1007/s11042-020-10073-7`.

[3] Seki, H., T. Hori, S. Watanabe, J. L. Roux, and J. R. Hershey: *End-to-End Multilingual Multi-Speaker Speech Recognition*. In *Proc. Interspeech 2019*, pp. 3755–3759. 2019. doi:10.21437/Interspeech.2019-3038. URL `http://dx.doi.org/10.21437/Interspeech.2019-3038`.

[4] Hannun, A., C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Ng: *DeepSpeech: Scaling up end-to-end speech recognition*. 2014.

[5] et al, A.: *Deep Speech 2: End-to-End Speech Recognition in English and Mandarin*. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 173–182. JMLR.org, 2016. Event-place: New York, NY, USA.

[6] Goodfellow, I., Y. Bengio, and A. Courville: *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[7] Bahar, P., T. Bieschke, and H. Ney: *A comparative study on end-to-end speech to text translation*. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 792–799. Sentosa, Singapore, 2019.

[8] Mirsamadi, S. and J. Hansen: *A study on deep neural network acoustic model adaptation for robust far-field speech recognition*. In *INTERSPEECH*. 2015.

[9] Sainath, T., R. Pang, D. Rybach, Y. He, R. Prabhavalkar, W. Li, M. Visontai, T. Strohman, Y. Wu, I. McGraw, and C.-C. Chiu: *Two-Pass End-to-End Speech Recognition*. pp. 2773–2777. 2019. doi:10.21437/Interspeech.2019-1341.

[10] Zeineldeen, M., A. Zeyer, R. Schlüter, and H. Ney: *Layer-normalized LSTM for Hybrid-HMM and End-to-End ASR*. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 7679–7683. Barcelona, Spain, 2020. URL `https://publications.rwth-aachen.de/record/795193`.

[11] Park, D., W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. Cubuk, and Q. Le: *SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition*. pp. 2613–2617. 2019. doi:10.21437/Interspeech.2019-2680.

[12] Hochreiter, S. and J. Schmidhuber: *Long short-term memory*. *Neural Comput.*, 9(8), p. 1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

[13] Graves, A., S. Fernández, F. Gomez, and J. Schmidhuber: *Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks*. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, p. 369–376. Association for Computing Machinery, New York, NY, USA, 2006. doi:10.1145/1143844.1143891. URL `https://doi.org/10.1145/1143844.1143891`.

[14] Shorten, C. and T. Khoshgoftaar: *A survey on image data augmentation for deep learning*. *Journal of Big Data*, 6, pp. 1–48, 2019.

[15] Park, D. S., W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le: *SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition*. In *Proc. Interspeech 2019*, pp. 2613–2617. 2019. doi:10.21437/Interspeech.2019-2680. URL `http://dx.doi.org/10.21437/Interspeech.2019-2680`.

[16] Heafield, K., I. Pouzyrevsky, J. H. Clark, and P. Koehn: *Scalable modified Kneser-Ney language model estimation*. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 690–696. Association for Computational Linguistics, Sofia, Bulgaria, 2013. URL `https://www.aclweb.org/anthology/P13-2121`.

[17] Heafield, K.: *KenLM: Faster and smaller language model queries*. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pp. 187–197. Association for Computational Linguistics, Edinburgh, Scotland, 2011. URL `https://www.aclweb.org/anthology/W11-2123`.

[18] Ardila, R., M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber: *Common voice: A massively-multilingual speech corpus*. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pp. 4211–4215. 2020.

[19] Panayotov, V., G. Chen, D. Povey, and S. Khudanpur: *Librispeech: An asr corpus based on public domain audio books*. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210. 2015. doi:10.1109/ICASSP.2015.7178964.

[20] Godfrey, J. J., E. C. Holliman, and J. McDaniel: *Switchboard: Telephone speech corpus for research and development*. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'92, p. 517–520. IEEE Computer Society, USA, 1992.

[21] Kingma, D. P. and J. Ba: *Adam: A method for stochastic optimization*. In Y. Bengio and Y. LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015. URL `http://arxiv.org/abs/1412.6980`.

Appendix Further Experiments

The following figures show the results of conducted 224 experiments on the proposed method with different learning rates. Figure 4 indicates that having a high learning rate lowers the performance once the first layer is allowed to be trained.

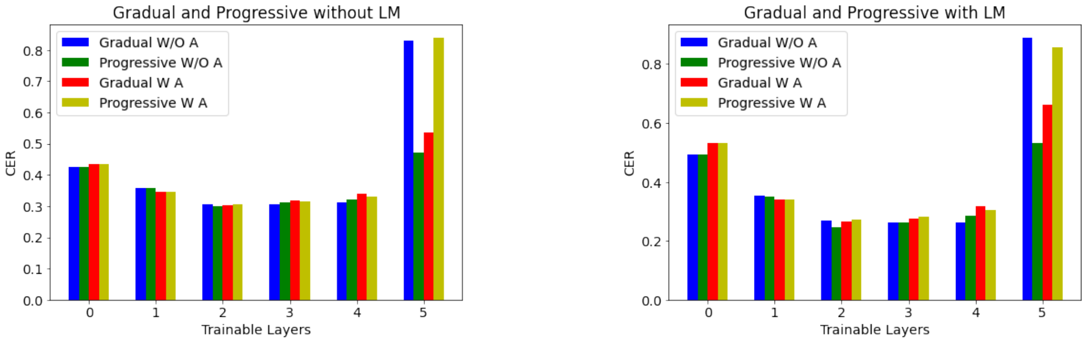

Figure 4 – Learning Rate 0.001 - Test Results for Progressive and Gradual with/with out Language Model and Augmentation. Individual numbers on trainable layers represent the allowed layers during training in reverse order.

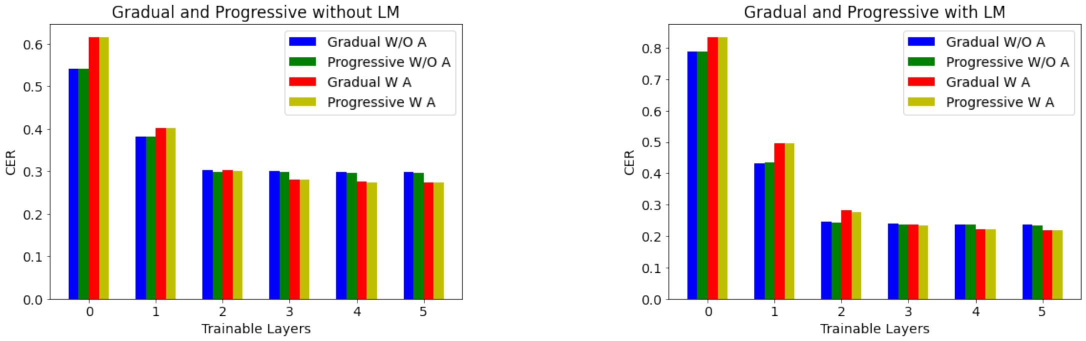

Figure 5 – Learning Rate 0.00001 - Test Results for Progressive and Gradual with/with out Language Model and Augmentation. Individual numbers on trainable layers represent the allowed layers during training in reverse order.

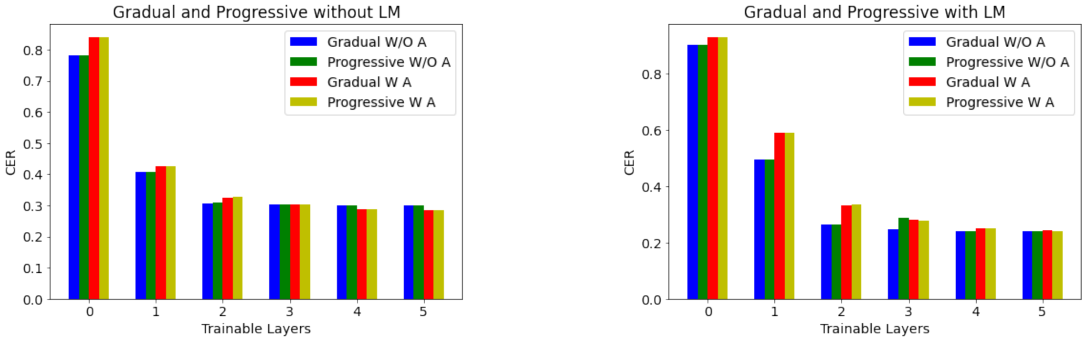

Figure 6 – Learning Rate 0.00005 - Test Results for Progressive and Gradual with/with out Language Model and Augmentation. Individual numbers on trainable layers represent the allowed layers during training in reverse order.