# Efficient Language Model Training through Cross-Lingual and Progressive Transfer Learning

**Malte Ostendorff**
DFKI GmbH
Berlin, Germany
malte.ostendorff@dfki.de

**Georg Rehm**
DFKI GmbH
Berlin, Germany
georg.rehm@dfki.de

## Abstract

Most Transformer language models are primarily pretrained on English text, limiting their use for other languages. As the model sizes grow, the performance gap between English and other languages with fewer compute and data resources increases even further. Consequently, more resource-efficient training methods are needed to bridge the gap for languages with fewer resources available. To address this problem, we introduce a cross-lingual and progressive transfer learning approach, called CLP-Transfer, that transfers models from a source language, for which pretrained models are publicly available, like English, to a new target language. As opposed to prior work, which focused on the cross-lingual transfer between two languages, we extend the transfer to the model size. Given a pretrained model in a source language, we aim for a same-sized model in a target language. Instead of training a model from scratch, we exploit a smaller model that is in the target language but requires much fewer resources. Both small and source models are then used to initialize the token embeddings of the larger model based on the overlapping vocabulary of the source and target language. All remaining weights are reused from the model in the source language. This approach outperforms the sole cross-lingual transfer and can save up to 80% of the training steps compared to the random initialization.

## 1 Introduction

Large language models based on the Transformer architecture (Vaswani et al., 2017) dominate today's NLP. These models are typically pretrained on primarily English text (Brown et al., 2020; Zhang et al., 2022; Black et al., 2022), except for a few multilingual models (Scao et al., 2022; Lin et al., 2021; Shliazhko et al., 2022). Given that multilingual models have been shown to perform suboptimal compared to monolingual ones (Conneau et al., 2020; Nozza et al., 2020), other languages than English benefit less from the recent progress in NLP. As the model sizes grow, the performance gap between the models for English and other languages with fewer resources increases even further. This gap is emphasized by Hoffmann et al. (2022), as they show that model performance is not only bound by computing resources but mainly by data. Consequently, more resource-efficient training methods are needed to bridge the gap for languages with fewer resources available.

Transfer learning is generally known to improve the training efficiency of various machine learning problems (Tan et al., 2018; Sun et al., 2018; Houlsby et al., 2019). In the context of language models, efficient methods for task, language, or domain adaption have been proposed, e.g., adapters (Chronopoulou et al., 2021; Pfeiffer et al., 2020), bias term training (Zaken et al., 2021), and prompt tuning (Guo et al., 2022). To obtain monolingual language models for low-resource languages, Minixhofer et al. (2021) and de Vries and Nissim (2021) have shown that available pretrained models, e.g., in English, can be recycled. These cross-lingual transfer learning approaches reduce the training effort. However, they only transfer across languages and neglect the sizes of the language models. While training a large model may not be feasible in a low-resource setting, training a small or medium model is likely possible, as demonstrated by AraGPT2 (Antoun et al., 2021), CamemBERT (Martin et al., 2020), GPT-fr (Simoulin and Crabbé, 2021), GBERT (Chan et al., 2020), or Finnish BERT (Virtanen et al., 2019).

This paper presents CLP-Transfer, which is a cross-lingual and progressive transfer learning approach for language models. As opposed to prior work, which focused on the cross-lingual transfer between two languages, we extend the transfer to the dimension of the model size. Given a large and pretrained model in a source language, we aim for a same-sized model in a target language. Instead of training the large model in the large language from scratch, we first train a smaller model that requires much fewer resources (or reuse a publicly available small model). Both small and source models are then used to initialize the token embeddings of the large target model based on the overlapping vocabulary of the source and target language. All remaining Transformer weights are reused from the large model in the source language.

We evaluate CLP-Transfer for decoder-only language models based on GPT2 (Radford et al., 2019) and BLOOM (Scao et al., 2022). We use German as the target language.[1] The source models are either in English or multilingual. Our experimental findings suggest that our approach outperforms the sole cross-lingual transfer and can save up to 80% of the training steps compared to the random initialization.

## 2 Related Work

**Cross-lingual Transfer.** Exploiting pretrained models or data across languages is a common approach in NLP research (Zoph et al., 2016; Lin et al., 2019; Nguyen and Chiang, 2017). One challenge when transferring a model to a new language is preserving the capabilities of the source language. Artetxe et al. (2020) proposed to replace the tokenizer and only train the token embeddings while freezing other Transformer layers of a multilingual BERT model. Such a transfer approach produces monolingual models that can be independently fine-tuned to specific languages. de Vries and Nissim (2021) followed a similar approach to transfer a GPT2 model to a new language. Specifically, they transfer English GPT2 to Dutch and Italian by exclusively relearning the token embeddings and not the other model weights. This forces the language model to learn token embeddings that are aligned between English and the target language. However, freezing most parameters also limits the model's ability to learn about the new language. More recently, Minixhofer et al. (2021) introduced the WECHSEL method that uses bilingual dictionaries to map the token embeddings from the source to the target language. It reuses the Transformer weights from the source model and continues training them. WECHSEL has been shown to outperform the transfer method from de Vries and Nissim.

**Progressive Transfer.** Going from a small to a larger model is also known as progressive growing and was originally proposed to improve training stability. Simonyan and Zisserman (2014) showed that starting from an efficient and small model and gradually increasing the model capacity yields more stable training. The paradigm of progressive growth can also be used to accelerate model training which has been shown for various model architectures. Karras et al. (2017) demonstrate this for GANs, Graves (2016) for RNNs, and Gu et al. (2021) for BERT language models. Furthermore, Gong et al. (2019) grow a BERT model in terms of its depth, i.e., they use trained weights of a shallow model to initialize a deeper model and achieve 25% shorter training time.

## 3 Methodology

This section introduces the methodology that we follow for efficient language model training.

Our objective is to obtain a large language model $M_t^{(\text{large})}$ with $p^{(\text{large})}$ parameters for a target language $t$. To increase the training efficiency, we omit the standard from-scratch training approach, i.e., randomly initializing the weights of $M_t^{(\text{large})}$. Instead, our goal is to find a good initialization of the parameter weights of $M_t^{(\text{large})}$ such that training effort is reduced. To achieve this, we exploit an already pretrained large language model $M_s^{(\text{large})}$, also with $p^{(\text{large})}$ parameters and the same model architecture but in a source language $s$, and a small pretrained language model $M_t^{(\text{small})}$, with significantly fewer parameters $p^{(\text{small})} << p^{(\text{large})}$ in the target language $t$. The Transformer layer weights $\boldsymbol{W}_t$ from the large target model are initialized with the weights of $M_s^{(\text{large})}$. Similarly, token embedding weights $\boldsymbol{V}_t$ for that the tokens that exist in both the target and source language vocabulary are initialized with $\boldsymbol{V}_s$. For the remaining token embeddings weights, a combination of $M_s^{(\text{large})}$ and $M_t^{(\text{small})}$ is used. To get our approach to work, we rely on two assumptions about the vocabularies and token embedding spaces of the source and target language models.

### 3.1 Assumptions

Our approach makes the following assumptions:

**Shared vocabulary.** Our approach relies on the tokenizers of source and target languages sharing

[1] German is not typically considered a low-resource language. However, at the time of writing, there are no monolingual German language models larger than 1B parameters publicly available. We will investigate other languages in the next iteration of this paper.

a substantial fraction of their vocabulary. Given the tokenizer in the source and target language with their vocabularies $V_s$ and $V_s$, we assume that the number of tokens occurring in both vocabularies $V_s \cap V_s$ is significantly larger than zero, i.e., $|V_s \cap V_s| >> 0$.[2] Languages with the same script and from the same language family typically share more tokens. For example, the overlap between German and English is higher compared to Arabic and English. Notably, there will be always a certain overlap since Byte-Pair Encoding (Sennrich et al., 2016) is the tokenization algorithm. As shown in Tab. 1, our assumption holds for the source and target combinations tested in this paper. While the English and German tokenizers share 24% of their vocabulary, the multilingual BLOOM tokenizer also shares 5% of the German vocabulary despite its much larger vocabulary size.

| **Vocabulary** $s$ | **Vocabulary** $t$ | $\lvert V_s \cap V_t \rvert$ |
|---|---|---|
| English GPT2 | German (ours) | 24.04% |
| Multilingual BLOOM | German (ours) | 5.55% |
| Multilingual XGLM | German (ours) | 2.62% |
| English GPT2 | Arabic GPT2 | 6.95% |
| English GPT2 | Finnish GPT2 | 13.71% |
| Multilingual BLOOM | Arabic GPT2 | 6.52% |
| Multilingual BLOOM | Finnish GPT2 | 3.54% |

Table 1: Number of overlapping vocabulary tokens between different tokenizers, normalized by the source vocabulary size. The tokenizers are English GPT2 (Radford et al., 2019), Arabic GPT2 (Antoun et al., 2021), Finnish GPT2[3], multilingual BLOOM (Scao et al., 2022), multilingual XGLM (Lin et al., 2021), and our German tokenizer.

**Token embeddings.** A language model has the token embeddings $\boldsymbol{V} \in \mathbb{R}^{|V| \times h}$ that map each token $v$ in the vocabulary $V$ to its vector representation $\boldsymbol{v} \in \mathbb{R}^h$ with the hidden size of $h$. For larger models, the hidden size $h$ of the token embedding is typically also larger compared to one of smaller models, i.e., $h^{(\text{large})} > h^{(\text{small})}$. Despite varying in terms of $h$, we assume that relative positioning in the token embedding space remains comparable across model sizes. The embeddings of a small model $\boldsymbol{V}^{(\text{small})}$ would share spacial properties with the embeddings $\boldsymbol{V}^{(\text{large})}$ of a large model.

To test this assumption, we compare token embeddings with different sizes from English OPT models (Zhang et al., 2022). Specifically, we retrieve the set of $k$-nearest neighbors $N_v$ with

[2]Another assumption is that tokenizers are identical across different model sizes as long as the language remains the same.

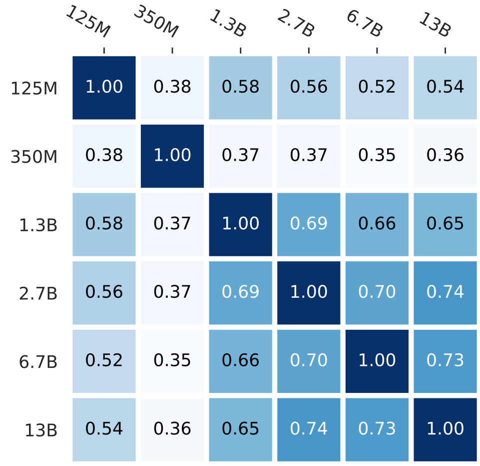


Figure 1: Similarity of token embeddings of different OPT model sizes measured as overlapping $k = 10$ nearest neighbors for all tokens in the vocabulary.

$k = 10$ for each token $v$ and measure the overlapping neighbors for different model sizes, e.g., $N_v^{(\text{large})} \cap N_v^{(\text{small})}$. This measure is normalized and computed for all available tokens.

As shown in Fig. 1, OPT token embeddings are comparable across model sizes. The similarity between embedding spaces increases when the model size is comparable. We find even between the smallest and the largest model (125M and 13B parameters) a 54% overlap. It is unclear why the 350M model has the lowest embedding similarity compared to all other models, independent of their size difference.

### 3.2 Cross-lingual & Progressive Transfer

The weights of a language model in a language $i$ are comprised of token embeddings $\boldsymbol{V}_i$ and the Transformer weights $\boldsymbol{W}_i$. We want to initialize $\boldsymbol{V}_t^{(large)}$ and $\boldsymbol{W}_t^{(large)}$ for our target language $t$ and the large model size. The Transformer weights are simply initialized with the ones from the source language $s$:

$$\boldsymbol{W}_t^{(large)} = \boldsymbol{W}_s^{(large)} \tag{1}$$

To initialize $\boldsymbol{V}_t^{(large)}$, we exploit $\boldsymbol{V}_s^{(large)}$ and $\boldsymbol{V}_t^{(small)}$, which are the token embeddings of a smaller model in the target language. The embeddings of overlapping tokens that simultaneously exist in the source and target vocabulary $v \in V_s \cap V_t$ are directly initialized with the source embeddings:

$$\boldsymbol{v_t} = \boldsymbol{v_s} \quad \text{if} \quad v \in V_s \cap V_t \tag{2}$$

When a token is not part of the overlapping vocabulary $v \notin V_s \cap V_t$, we initialize its embedding $\boldsymbol{v_t}$ as the weighted average over the embeddings $\hat{\boldsymbol{v}}$ of the overlapping token:

$$\boldsymbol{v}_{\boldsymbol{t}}^{(\text{large})} = \sum_{\hat{v} \in V_s \cap V_t} \frac{\hat{\boldsymbol{v}}_s^{(\text{large})}}{\delta\left(v_t, \hat{v}_t\right)} \quad \text{if} \quad v \notin V_s \cap V_t \tag{3}$$

The weight function $\delta$ has the objective to transfer the spacial properties from the small model to the large model and is defined as the normalized cosine similarity of the small embeddings of overlapping $v$ and missing $\hat{v}$ tokens:

$$\delta(v, \hat{v}) = \frac{cos\left(\boldsymbol{v}_t^{(small)}, \hat{\boldsymbol{v}}_t^{(small)}\right)}{\sum\limits_{\substack{\hat{v}' \in V_s \cap V_t, \\ v' \in V_s \cup V_t}} cos\left(\boldsymbol{v}_t'^{(small)}, \hat{\boldsymbol{v}}_t'^{(small)}\right)} \tag{4}$$

The intuition is those embeddings that are more similar in the $\boldsymbol{E}_t^{(small)}$ should contribute more to the contruction of their corresponding token in the large model. This approach allows us to recycle the pretrained weights of a source large model while preserving the spacial properties of the embedding space of the target language and simultaneously adjusting it to the vocabulary of our target language.

## 4 Experiment Design

In the experiments, we evaluate the CLP-Transfer approach by transferring the English GPT2 (Radford et al., 2019) and multilingual BLOOM (Scao et al., 2022) to a monolingual German language model. Both model types are evaluated at different scales. More specifically, we grow the GPT2 model from 117M to 1.5B parameters and the BLOOM model from 1.5B to 6.4B parameters.

### 4.1 Models

**Model Archiectures.** Both models (GPT2 and BLOOM) are decoder-only language models based on the Transformer architecture (Vaswani et al., 2017) and are trained with the causal language modeling objective. GPT2 uses learned positional embeddings, whereas BLOOM uses ALiBi (Press et al., 2022). Another difference is that BLOOM applies normalization on the token embedding layer to improve training stability.

In our experiments with GPT2, we aim for a monolingual German model with the size of GPT2-XL with 1.5B parameters. The source model is the English GPT2-XL as provided by Radford et al. (2019)[4] with 48 layers, 25 attention heads, and a hidden size of 1600. As small German model, we use a GPT2-base model with 117M parameters trained with WECHSEL (Minixhofer et al., 2021).[5] The small German model has 12 layers, 12 attention heads, and a hidden size of 768.

To test our transfer method for another model type and size, we also conduct experiments with BLOOM. For this experiment, our objective is the training of a German model based on BLOOM with 7.1B parameters as the source model.[6]. The BLOOM 7.1B model has 30 layers, 32 attention heads, and a hidden size of 4096. Our German BLOOM target model uses a different tokenizer with a smaller vocabulary size (see below). Therefore, its token embedding layer contains fewer parameters than the multilingual BLOOM model. As a result, the target model has only 6.4B parameters. The small German model is a BLOOM model with 1.5B parameters trained with our method (24 layers, 16 attention heads, and a hidden size of 2048).[7]

**Tokenizers.** All used tokenizers are based on Byte-Pair Encoding (Sennrich et al., 2016). The vocabulary size of English GPT2 is 50,257 tokens. BLOOM is multilingual and covers a diverse set of 46 natural languages and 13 programming languages. Therefore, BLOOM's vocabulary size is 250,880 tokens, five times larger than the one from English GPT2. For our German tokenizer, we opt for the same vocabulary sizes as the English one (50,257 tokens) and train it on the German subset of OSCAR v2019 (Ortiz Su'arez et al., 2019).

### 4.2 Datasets

The GPT2 and BLOOM models are trained with two different datasets.

**GPT2 Training.** The German GPT2-CLP training relies exclusively on Web-crawled data from the German subset of OSCAR v2019 (Ortiz Su'arez et al., 2019).[8] We follow the methodology from Minixhofer et al. (2021) to construct a separate

[4] https://hf.co/gpt2-xl
[5] https://hf.co/malteos/gpt2-wechsel-german-ds-meg
[6] https://hf.co/bigscience/bloom-7b1
[7] https://hf.co/malteos/bloom-1b5-clp-german
[8] https://hf.co/datasets/oscar (subset: `unshuffled_deduplicated_de`)

training and validation dataset. Specifically, we used the first 4 GB of OSCAR as the training dataset, then the next 0.4GB as the validation dataset. The GPT2 training dataset comprises approximately 30.8B tokens.

**BLOOM Training.** To train the German BLOOM-CLP 6.4B model, we construct another dataset. We use again Web-crawled content from the German subset OSCAR but the more recent version of v22.01 (excluding content tagged as *header*, *footer*, *noisy*, or *adult*) and from the GC4 Corpus[9] (including only the *head* and *middle* parts). As both data sources originate from CommonCrawl and potentially have duplicated content, we deduplicate the Web-crawled content using the approach from Lee et al. (2022). We complement the Web-crawled content with German court decisions from Open Legal Data (Ostendorff et al., 2020). The BLOOM training dataset comprises approximately 50.4B tokens.

**Evaluation Datasets.** We evaluate the models for their language modeling ability using the OSCAR validation set from the GPT2 training[10], and for zero-shot learning on German downstream tasks. The tasks are sentiment analysis from GermEval 2017 (Wojatzki et al., 2017), hate speech classification from GermEval 2018 (Wiegand and Siegel, 2018), news topic classification from GNAD10 (Schabus et al., 2017), paraphrase identification from PAWSX (Yang et al., 2019), natural language inference from XNLI (Conneau et al., 2018), and stance detection from X-Stance (Vamvas and Sennrich, 2020). All evaluation tasks are implemented using the `lm-evaluation-harness` framework (Gao et al., 2021).[11]

### 4.3 Baselines

We compare against the following baselines:

**From-Scratch Training.** The language model is trained from scratch in the target language with randomly initialized weights.

The from-scratch baseline for the BLOOM experiments (BLOOM 6.7B) was trained with minor changes to the transferred BLOOM-CLP 6.4B model. The baseline BLOOM 6.7B follows the model size proposed by Brown et al. (2020). It has 32 layers instead of 30 layers and was not trained on GC4 and Open Legal Data but on other German datasets.

**WECHSEL.** The WECHSEL method as introduced by Minixhofer et al. (2021) applies cross-lingual transfer to monolingual language models. WECHSEL uses bilingual dictionaries to map the token embeddings from a source language to a target language and reuses the Transformer weights from the source model. WECHSEL has been shown to outperform the transfer method from de Vries and Nissim (2021).

**Multilingual Models.** Lastly, we compare the monolingual German models against multilingual models trained on German data. We evaluate XGLM (Lin et al., 2021) ranging from 564M to 7.5B parameters and mGPT (Shliazhko et al., 2022) with 1.3B parameters. XGLM was trained on approx. 5.4% German data and mGPT on 8.2% German data.

## 5 Results

We show the training results of two monolingual German language models, i.e., GPT2 1.5B and BLOOM 6.2B. The models are evaluated regarding their training efficiency and downstream task performance.

### 5.1 Transfering GPT2

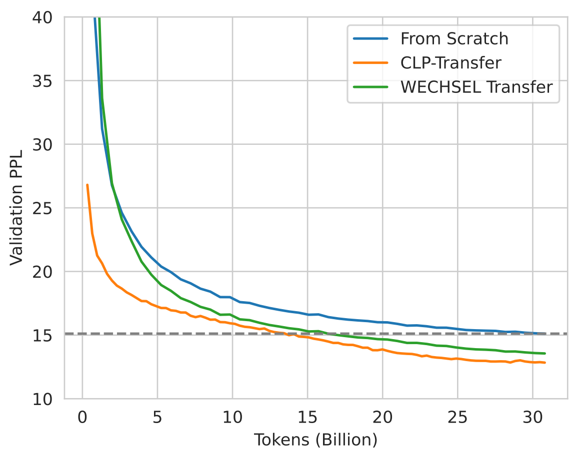


Figure 2: GPT2-XL German (1.5B parameters). Validation perplexity w.r.t. the number of consumed tokens comparing from-scratch training (random initialization), WECHSEL (cross-lingual transfer), and our CLP-Transfer approach. CLP-Transfer achieves the same PPL as from-scratch training but already after 50% of tokens (dashed line).

[9] https://german-nlp-group.github.io/projects/gc4-corpus.html

[10] https://hf.co/datasets/malteos/wechsel_de

[11] https://github.com/OpenGPTX/lm-evaluation-harness

The goal of our first experiment is the training of a German GPT2-XL model with 1.5B parameters. CLP-Transfer is compared against from-scratch training and the cross-lingual transfer method from WECHSEL. Fig. 2 shows the validation perplexity (PPL) of each method in relation to the training progress measured in consumed tokens.

We find that CLP-Transfer outperforms the baselines. The validation PPL of CLP-Transfer is constantly the lowest of all three methods. At the end of the training (after 30.8B tokens), CLP-Transfer yields a 12.8 PPL, followed by WECHSEL with 13.5 PPL. The worst result has the from-scratch training with 15.1 PPL. CLP-Transfer achieve the same PPL as from-scratch training but already have 50% of the consumed tokens. During the first phase of the training (0-5B tokens), the improvements of CLP-Transfer are most significant. These results demonstrate that our transfer learning approach is superior to from-scratch training even at the end of the training or can achieve the same results more efficiently. Moreover, the additional use of a small model in the target language yields further efficiency gains compared to the sole cross-lingual transfer done by WECHSEL.

### 5.2 Transfering BLOOM

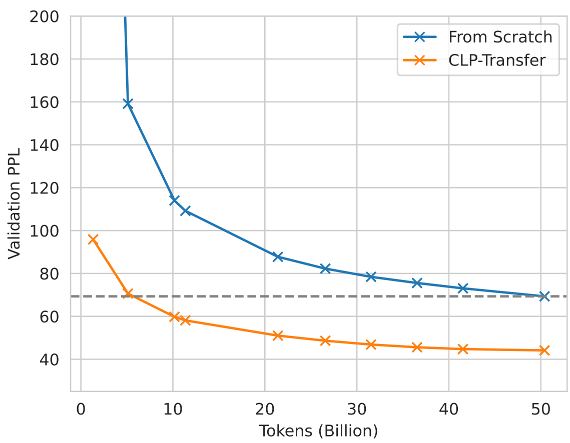


Figure 3: BLOOM-6B-German. Validation perplexity w.r.t. the number of tokens comparing from-scratch training (random initialization) and our CLP-Transfer approach. CLP-Transfer achieves the same PPL as from-scratch training but after 20% of tokens (dashed line).

The second experiment applies CLP-Transfer on a multilingual BLOOM model[6] to train a monolingual German model with 6.4B parameters. In this experiment, we compare only against from-scratch training. We discarded WECHSEL from this experiment due to the results from the first experiment and due to WECHSEL being made for the transfer of monolingual models and not multilingual ones.

As shown in Fig. 3, CLP-Transfer again outperforms the from-scratch training. After complete training on 50.4B tokens, CLP-Transfer yields a 44.1 PPL, whereas from-scratch training is significantly worse with 69.3 PPL. 20% of training tokens are sufficient for CLP-Transfer to be on par with from-scratch training. This suggests that the efficiency gains from CLP-Transfer are even more prevalent at 6B compared to 1.5B parameters. We attribute this outcome to the training data containing too few tokens for 6B models. The validation PPL is still decreasing at the end of the training suggesting that the model is not fully trained yet. According to Hoffmann et al. (2022), a compute-optimal language model at the 6B scale would require approx. 142B tokens which our BLOOM model training did not consume. The GPT2 training is much closer to being compute-optimal (Hoffmann et al. suggest 33B tokens for a 1.5B model).

### 5.3 Downstream Tasks

Even though we trained the models exclusively with a causal language modeling objective, we want them to perform well on other downstream tasks, as shown by Brown et al. (2020). Hence, we compare the models and additional baselines on six German benchmarks in a zero-shot setting. Given that the from-scratch trained BLOOM 6.7B model (50B tokens) is presumable under-trained, we add an additional variation that was trained on 22B more tokens, i.e., BLOOM 6.7B (72B tokens). The evaluated tasks are sentiment analysis (GermEval 2017), hate speech classification (GermEval 2018), news topic classification (GNAD10), paraphrase identification (PAWSX), natural language inference (XNLI), and stance detection (X-Stance). Tab. 2 reports the validation PPL on German OSCAR[10], the results for the individual tasks, and the average over the tasks.

The zero-shot performance of all models is disappointing. Most models achieve results on par or worse than the random baseline. Only the largest models (more than 6B parameters) are better than the random baseline on average. The BLOOM-CLP 6.4B model has the best average score of 0.43, followed by the from-scratch trained BLOOM 6.7B (72B tokens) and XGLM 7.5B.

We hypothesize that this outcome is due to the

Table 2: Evaluation results of German downstream tasks in a zero-shot setting. The average score excludes the OSCAR validation perplexity (PPL). Smaller models are on par or worse than the random baseline. Our transfer model BLOOM-CLP 6.4B achieves the best results on average.

| **Task** → <br> **Model** ↓ / **Metric** → | **Oscar** <br> PPL (↓) | **GEval17** <br> F1 (↑) | **GEval18** <br> F1 (↑) | **GNAD10** <br> F1 (↑) | **PAWSX** <br> F1 (↑) | **XNLI** <br> Acc. (↑) | **XStance** <br> F1 (↑) | **Avg.** <br> (↑) |
|---|---|---|---|---|---|---|---|---|
| Random | - | 0.33 | 0.50 | 0.11 | 0.50 | 0.33 | 0.50 | 0.38 |
| *Multilingual models:* | | | | | | | | |
| mGPT 1.3B | 2274.80 | 0.36 | 0.51 | 0.08 | 0.49 | 0.37 | 0.49 | 0.38 |
| XGLM 564M | 179.59 | 0.05 | 0.40 | 0.05 | 0.46 | 0.44 | 0.50 | 0.32 |
| XGLM 1.7B | 105.10 | 0.04 | 0.35 | **0.19** | **0.58** | 0.45 | 0.40 | 0.34 |
| XGLM 7.5B | 66.74 | 0.51 | 0.51 | 0.06 | 0.50 | 0.39 | 0.41 | 0.40 |
| *Monolingual German models:* | | | | | | | | |
| GPT2-WECHSEL 117M | 594.40 | 0.04 | 0.51 | 0.18 | 0.49 | 0.40 | **0.51** | 0.35 |
| GPT2-XL-WECHSEL 1.5B | 157.95 | 0.05 | **0.55** | 0.10 | 0.41 | **0.49** | 0.34 | 0.32 |
| GPT2-XL-CLP 1.5B | 46.33 | 0.05 | 0.02 | 0.07 | 0.46 | **0.49** | 0.34 | 0.24 |
| GPT2-XL 1.5B from scratch | 187.71 | 0.04 | 0.51 | 0.15 | 0.52 | 0.47 | 0.34 | 0.34 |
| BLOOM-CLP 1.5B | 49.80 | 0.04 | 0.14 | 0.11 | 0.44 | 0.48 | 0.38 | 0.26 |
| BLOOM-CLP 6.4B (50B tokens) | **44.09** | **0.56** | 0.51 | 0.13 | 0.52 | 0.43 | 0.44 | **0.43** |
| BLOOM 6.7B from scratch (50B tokens) | 69.32 | 0.51 | 0.52 | 0.13 | 0.41 | 0.38 | 0.42 | 0.39 |
| BLOOM 6.7B from scratch (72B tokens) | 64.03 | **0.56** | 0.51 | 0.09 | 0.40 | 0.37 | 0.49 | 0.40 |

model size and token count being still too small. Studies from Black et al. (2022) or Shliazhko et al. (2022) report similar near-random results for models with comparable sizes. Another reason might be the poorly translated test datasets that produce less meaningful results. For instance, PAWSX contains a large fraction of machine-translated samples. To improve the downstream task performance, promising approaches are prompt engineering, i.e., tailoring the prompts more to the German language, and multi-task fine-tuning, as demonstrated by BLOOMZ (Muennighoff et al., 2022) or FLAN (Wei et al., 2022).

## 6 Conclusion

This paper introduces CLP-Transfer, which is a cross-lingual and progressive transfer learning approach for the efficient training of large language models. Our experiments demonstrate that monolingual German language models initialized with CLP-Transfer reduce the training effort. The CLP-Transfer models achieve better results when trained on the same number of tokens than from-scratch training or WECHSEL transfer. To obtain the same perplexity as from-scratch training, CLP-Transfer needs only 50% (GPT-2) or even 20% (BLOOM) of the original token count. This corresponds to a 50% or 80% reduction in training effort. Such a reduction lowers the barriers to the training of large language models in low-resource settings.

The training efficiency is achieved by exploiting publicly available models, i.e., English or multilingual large models and small models in the target language. CLP-Transfer relies on the assumptions that vocabularies of source and target language have significant overlap and that small and large model have similar token embedding spaces.

We make the pretrained model checkpoints and our source code publicly available on HuggingFace[12] and GitHub[13]. Furthermore, we provide a Web-based demo in which the German BLOOM-CLP model with 6.4B parameters can be prompted.[14]


## Acknowledgements

The work presented in this paper has received funding from the German Federal Ministry for Economic Affairs and Climate Action (BMWK) through the project OpenGPT-X (project no. 68GX21007D).

The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. and the GWK support for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCSSupercomputer JUWELS at Jülich Supercomputing Centre (JSC) and through the Center for Information Services and HPC (ZIH) at TU Dresden.


[12] https://huggingface.co/malteos/bloom-6b4-clp-german
[13] https://github.com/malteos/clp-transfer
[14] https://ostendorff.org/clp

## References


Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An Open-Source Autoregressive Language Model.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020-Decem.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexandra Chronopoulou, Matthew E. Peters, and Jesse Dodge. 2021. Efficient Hierarchical Domain Adaptation for Pretrained Language Models.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Wietse de Vries and Malvina Nissim. 2021. As Good as New. How to Successfully Recycle English GPT-2 to Make Models for Other Languages. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 836–846.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Linyuan Gong, Di He, Zhuohan Li, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Efficient training of BERT by progressively stacking. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2337–2346. PMLR.

Alex Graves. 2016. Adaptive Computation Time for Recurrent Neural Networks.

Xiaotao Gu, Liyuan Liu, Hongkun Yu, Jing Li, Chen Chen, and Jiawei Han. 2021. On the Transformer Growth for Progressive BERT Training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5174–5180, Online. Association for Computational Linguistics.

Xu Guo, Boyang Albert Li, and Han Yu. 2022. Improving the sample efficiency of prompt tuning with domain adaptation. *ArXiv*, abs/2210.02952.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training Compute-Optimal Large Language Models. 3(2020):1–36.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive Growing of GANs for Improved Quality, Stability, and Variation.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2021. Few-shot Learning with Multilingual Language Models.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2021. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual Generalization through Multitask Finetuning.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [MASK]? Making Sense of Language-Specific BERT Models.

Pedro Javier Ortiz Su'arez, Benoit Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut f"ur Deutsche Sprache.

Malte Ostendorff, Till Blume, and Saskia Ostendorff. 2020. Towards an Open Platform for Legal Information. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 385–388.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A Framework for Adapting Transformers. pages 46–54.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *arXiv*.

Teven Le Scao, Thomas Wang, Daniel Hesslow, Lucile Saulnier, Stas Bekman, M Saiful Bari, Stella Biderman, Hady Elsahar, Niklas Muennighoff, Jason Phang, Ofir Press, Colin Raffel, Victor Sanh, Sheng Shen, Lintang Sutawika, Jaesung Tae, Zheng Xin Yong, Julien Launay, and Iz Beltagy. 2022. What Language Model to Train if You Have One Million GPU Hours?

Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One million posts: A data set of german online discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1241–1244, Tokyo, Japan.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mGPT: Few-Shot Learners Go Multilingual.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Antoine Simoulin and Benoit Crabbé. 2021. Un modèle Transformer Génératif Pré-entrainé pour le ______ français. In *Traitement Automatique des Langues Naturelles*, pages 246–255, Lille, France. ATALA.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. 2018. Meta-transfer learning for few-shot learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 403–412.

Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*.

Jannis Vamvas and Rico Sennrich. 2020. X-Stance: A multilingual multi-target dataset for stance detection. In *Proceedings of the 5th Swiss Text Analytics*

*Conference (SwissText) 16th Conference on Natural Language Processing (KONVENS)*, Zurich, Switzerland.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, (Nips):6000–6010.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners.

Michael Wiegand and Melanie Siegel. 2018. Overview of the germeval 2018 shared task on the identification of offensive language. In *Proceedings of the GermEval*.

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. Germeval 2017: Shared task on aspect-based sentiment in social media customer feedback. *Proceedings of the GermEval*, pages 1–12.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. (i).

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pretrained Transformer Language Models.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.