The Privileged Students: On the Value of Initialization in Multilingual Knowledge Distillation

Anonymous ACL submission

Abstract

Knowledge distillation (KD) has proven to be a successful strategy to improve the performance of smaller models in many NLP tasks. However, most of the work in KD only explores monolingual scenarios. In this paper, we investigate the value of KD in multilingual settings. We find the significance of KD and model initialization by analyzing how well the student model acquires multilingual knowledge from the teacher model. Our proposed method emphasizes copying the teacher model's weights directly to the student model to enhance initialization. Our findings show that model initialization using copy-weight from the fine-tuned teacher contributes the most compared to the distillation process itself across various multilingual settings. Furthermore, we demonstrate that efficient weight initialization preserves multilingual capabilities even in low-resource scenarios.

1 Introduction

Multilingual models face challenges related to both data availability and computational resources (Ruder, 2022; Nityasya et al., 2021; Aji et al., 2022), which makes producing lightweight multilingual models challenging. Recent work has proposed utilizing Knowledge Distillation (KD) in multilingual contexts (Ansell et al., 2023). However, due to data scarcity, questions arise about how to effectively transfer knowledge between teacher and student models.

The multilingual KD approach follows the training methodology of TinyBERT (Jiao et al., 2020), involves a two-step distillation process where a smaller student model is pre-trained before fine-tuning to acquire multilingual knowledge from a larger teacher model. However, this method requires substantial data for the pre-training step, thus increasing computational costs. We argue that this pre-training step is crucial to make the KD work in this multilingual scenario, as the scenario requires strong cross-lingual prior knowledge as an initialization to perform well, especially in low-resource scenarios (Conneau et al., 2020). This initialization serves as important inductive biases for downstream tasks across multiple languages.

To investigate this, we analyze the impact of both KD and model initialization on multilingual model performance. We propose an efficient initialization method that involves alternately copying weights from the teacher model to the student model, similar to the approach used in DistilBERT (Sanh et al., 2020). While weight copying has been employed previously, its effectiveness in multilingual scenarios remains underexplored. Our study aims to comprehensively examine how KD and weight copying affect multilingual model performance, learning speed, and cross-lingual generalization.

Our contributions to this work are as follows:

1. We demonstrate that fine-tuning a well-initialized model, specifically by copying the teacher model's weights, significantly impacts downstream task performance more than leveraging the distillation objective function across various multilingual settings.

2. We show that weight-copy enables zero-shot cross-lingual generalization, preserving multilingual capabilities even in low-resource scenarios.

3. Our experiments reveal that the weight-copy approach leads to faster learning speeds and exhibits multilingual knowledge. Among the weight-copying strategies we compared, directly copying the teacher's weights resulted in the fastest learning speed across different data sizes.

2 Methodology

Knowledge distillation (KD) is a technique used to transfer knowledge from a large, trained teacher

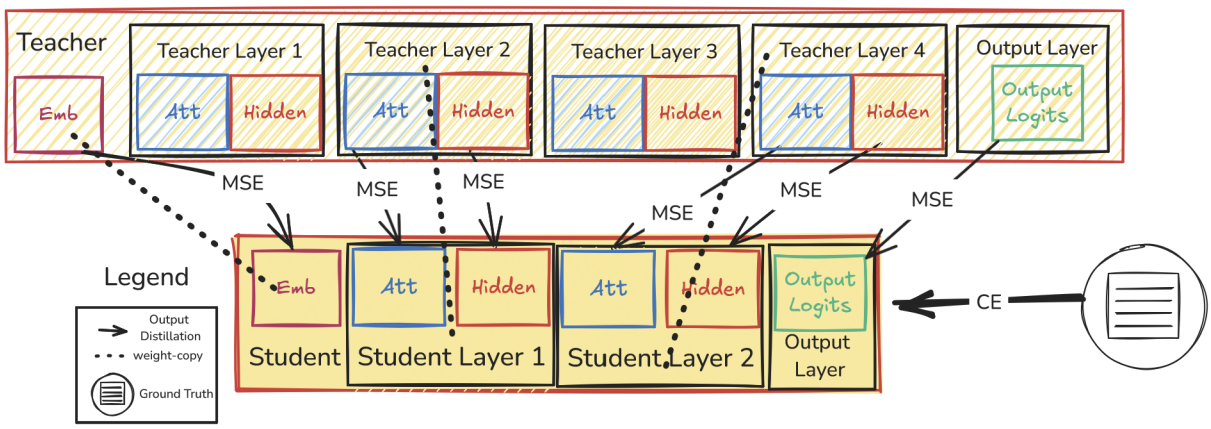

Figure 1: Overall architecture of Knowledge Distillation used in this paper where the teacher distills its knowledge using Mean Square Loss (MSE) and followed by Cross Entropy (CE) Loss with respect to the ground truth from the labeled dataset. This is an example of pair of student-teacher models that have 2 and 4 layers, respectively. Att denotes attention output, Hidden denotes hidden output, and Emb denotes Embedding weights.

model to a smaller student model. This process aims to retain the large model's performance while reducing the computational cost during inference. KD involves training the student model to mimic the outputs of the teacher model, often using a combination of the teacher's soft target outputs and the ground truth labels.

Various extensions of KD have been proposed to enhance its effectiveness. For instance, TinyBERT (Jiao et al., 2020; Ansell et al., 2023) introduces a two-step distillation process. First, the student model is pre-trained on a large corpus to acquire good initialization suitable for the next step. Afterward, the model is fine-tuned for the desired tasks. This approach requires substantial data and computational resources, making it challenging to implement with limited resources.

In this work, we aim to explore the impact of these components of knowledge distillation in multilingual settings, focusing on efficiency. Instead of the extensive pre-training step used in Jiao et al., 2020; Ansell et al., 2023, we employ a simpler and more efficient initialization approach by copying the weights from the teacher model to the student model, inspired by DistilBERT (Sanh et al., 2020).

2.1 Distillation Architecture

We utilize knowledge distillation (KD), comprised of a teacher T and a student S model. The student model always has fewer layers than the teacher model. We follow TinyBERT (Jiao et al., 2020)'s objective loss and architecture. The loss of the KD comprises embedding loss $\mathcal{L}_{embd}$, hidden-layer loss $\mathcal{L}_{hidn}$, attention loss $\mathcal{L}_{att}$, and prediction-layer loss $\mathcal{L}_{pred}$. These objective functions can be formulated as follows:

$$\mathcal{L}_{att} = \frac{1}{l}\frac{1}{h}\sum_{i=1}^{l}\sum_{j=1}^{h}\text{MSE}(A_S^i, A_T^k) \quad (1)$$

$$\mathcal{L}_{hid} = \frac{1}{l}\sum_{i=1}^{l}\text{MSE}(W \cdot H_S^i, H_T^k) \quad (2)$$

$$\mathcal{L}_{embd} = \text{MSE}(W \cdot E_S, E_T) \quad (3)$$

$$\mathcal{L}_{pred} = \text{MSE}(z_S, z_T) \quad (4)$$

Where A, H, E, and z are the values of the attention outputs, hidden layers' outputs, embedding layer's outputs, and the logits, respectively, for the teacher T or student S models. Indices of model layers and attention heads are denoted as l and h. If the student model's hidden unit dimension is smaller than the teacher's, we leverage a projection weight W to match the hidden unit dimension. Otherwise, W is an identity matrix[1].

The mapping function of student and teacher model is based on the best ablation results of Jiao et al. (2020), which defined as follows:

$$k = i \cdot \frac{N_T}{N_S}, \quad i \in [1, N_S], \quad k \in [1, N_T] \quad (5)$$

In this formula, i represents the index of the student's layer, while k is the index of the corresponding teacher's layer. N_S denotes the total number of

[1]In the implementation, we omit W instead.

Information	massive	tsm	universal-ner
Number of Training Data	11,514	1,839	6,366*
Number of Classes	60	3	7
Number of Languages	52	8	9
unseen lang partition	"am-ET", "cy-GB", "af-ZA", "km-KH", "sw-KE", "mn-MN", "tl-PH", "kn-IN", "te-IN", "sq-AL", "ur-PK", "az-AZ", "ml-IN", "ms-MY", "ca-ES", "sl-SL", "sv-SE", "ta-IN", "nl-NL", "it-IT", "he-IL", "pl-PL", "da-DK", "nb-NO", "ro-RO", "th-TH", "fa-IR"	"arabic", "french", "hindi", "portuguese"	"da-ddt", "pt-bosque", "sr-set", "sk-snk", "sv-talbanken"

Table 1: Data statistics for massive, tsm, and universal-ner. Each language consists of the same number of instances in both datasets, except universal-ner which number instance varies across languages. unseen lang denotes language subset used in the zero-shot cross-lingual experiment. The rest of the languages are categorized as seen lang. universal-ner data does not include partition that does not have training set. * denotes the mean of number of instances across languages.

layers in the student model, and N_T represents the total number of layers in the teacher model.

The KD loss can be formulated as follows:

$$\mathcal{L}_{\mathcal{KD}} = \mathcal{L}_{att} + \mathcal{L}_{hid} + \mathcal{L}_{embd} + \mathcal{L}_{pred}$$

We calculate the classification loss $\mathcal{L}_{clf}$ as follows:

$$\mathcal{L}_{clf} = CE(z^S, GT)$$

Where GT is the ground truth of the observed instance.

Finally, we obtain the overall loss $\mathcal{L}_{overall}$ which is going to be minimized in the training process:

$$\mathcal{L}_{overall} = \mathcal{L}_{\mathcal{KD}} + \mathcal{L}_{clf}$$

We use Mean Square Error (MSE) instead of KL Divergence due to faster convergence and higher performance, as supported by the experiment of Nityasya et al., 2022. The overall architecture can be seen in Figure 1.

2.2 Model Initialization

To avoid the expensive pre-training step used by Jiao et al., 2020; Ansell et al., 2023, we adopt the model initialization approach from DistilBERT (Sanh et al., 2020), where the student model's weights are initialized by copying the weights of the teacher model.

We alternately copy the weights of the teacher's embedding layer and classification layers to the student model. For the self-attention layer, we copy the weights based on the following mapping function:

$$\text{SA}_T^j = \text{SA}_S^{i*2} \quad \text{for} \quad i, j \in \mathbb{Z}^+ \tag{6}$$

Here, SA denotes the self-attention layers of the teacher T and student S models, respectively. The notations i and j indicate the indices of the student and teacher self-attention layers, respectively. To illustrate, the second self-attention layer of the teacher model will be mapped to the first self-attention layer of the student model.

If the student's hidden unit dimension is smaller than the teacher's, we follow the approach of Xu et al., 2024 by selecting evenly spaced elements in the teacher's linear weight and bias for the self-attention layer to map the student's self-attention layer correspondingly. For instance, suppose the teacher has a linear weight of 4x4, and the student has a 2x2 matrix; we select the 1st and 3rd slices along both the first and second dimensions. For the bias, we do the slicing in one dimension instead.

3 Experiment Setup

We provide three experiment setups: data, model, and training, that will be consistently used throughout this work.

Data We utilized massive (FitzGerald et al., 2022), Tweet Sentiment Multilingual dataset (denoted as tsm) (Barbieri et al., 2022), and universal-ner (Mayhew et al.). We selected these datasets to observe the behavior of multilingual performance under different situations: high-resource data with parallel data (massive) and low-resource data with non-parallel data (tsm). We also use universal-ner for more complex mul-

Model	Method	Layers	XLM-R			mDeBERTa			Avg
			massive	tsm	universal-ner	massive	tsm	universal-ner	
Teacher	from-base	12	80.18	70.10	87.66	81.36	66.96	88.81	79.18
Student	from-scratch	6	75.19	50.20	45.68	75.93	49.86	46.21	57.18
	from-scratch + KD	6	79.23	54.13	48.29	76.22	51.80	39.55	60.30
	from-teacher	6	81.18	62.99	79.50	80.45	58.12	78.20	73.41
	from-teacher + KD	6	**81.63**	**67.61**	**80.18**	**82.31**	**61.08**	**79.17**	**75.33**

Table 2: Comparative performance (F1-scores %) of XLM-R and mDeBERTa models across different datasets (all languages), initialization methods, and with/without knowledge distillation (KD). The teacher model has 12 layers, while all student models have 6 layers. All models are trained with all languages for each data. Bold denotes best methods for each data and their average.

tilingual task which has medium-resource data with non-parallel data. In this experiment, these data comprises of languages that are divided into `unseen lang` and `seen lang` to simulate zero-shot cross-lingual scenarios. Table 1 shows the corresponding datasets' data statistics and language partition. The detailed language partition information can be seen in Appendix B.

Model We used `transformers` library (Wolf et al., 2020) and the off-the-shelf implementation of `xlm-roberta` (Liu et al., 2019) and `mdeberta` (He et al., 2021) models. We used a reduction factor of 2 for the number of student layers compared to the teacher. Additionally, we compared performance by reducing the hidden units by half and keeping the hidden units the same as the teacher's. We experimented with three different model initialization scenarios: copying the weights from `xlm-roberta-base` or `mdeberta-v3-base` (`from-base`), copying from the fine-tuned teacher (`from-teacher`), and initializing without copying from any model (`from-scratch`). we compared their performances to understand the differences between these strategies.

Training To fine-tune the model and perform knowledge distillation, we used AdamW (Loshchilov and Hutter, 2019) as the optimizer, with the default hyperparameters stated in the `transformers` library. We set the number of epochs to 30 and obtained the best results evaluated on the development set using the F1 score metric. The evaluation steps for `massive` are as follows: 5000 steps for §4.1 and §4.2, and 100 steps for §4.3. For `tsm` and `universal-ner`, we set the evaluation steps to 500, 250, and 60 for §4.1, §4.2, and §4.3, respectively. These differences are due to the data sizes used in the corresponding experiments. The rest of the hyperparameters follow the default configuration in the `transformers` library. We used an A100 GPU to train our models to run our experiments, running each model's training three times with different seeds. Performances are evaluated using Macro F1-Score. For `universal-ner`, it is evaluated using `overall-f1` of `seqeval` wrapped in `evaluate` package[2].

4 Multilingual Transferability in KD

The ability of knowledge distillation (KD) to transfer knowledge across multiple languages efficiently remains unexplored. As mentioned in §2, two components need to be analyzed: model initialization and the distillation process itself. It is still unclear which of these factors contributes the most to the overall performance. Also, in multilingual scenarios, we often encounter situations where not all languages are covered in the training set. Understanding whether KD can facilitate zero-shot cross-lingual (ZSCL) generalization and effectively transfer multilingual knowledge remains unexplored. Building a multilingual dataset is tedious, thus leading researchers to opt for using one language. As a result, it is desirable if such setup can achieve cross-lingual generalization (Artetxe and Schwenk, 2019).

4.1 Weight Copy Transfers More Information vs Distillation Loss

Table 2 presents a comparative analysis of model performance with and without the copy-weight initialization strategy using all languages in the training dataset. The results demonstrate that the `from-teacher` approach significantly outper-

[2] https://huggingface.co/spaces/evaluate-metric/seqeval

Initialization	XLM-R			mDeBERTa			Avg
	massive	tsm	universal-ner	massive	tsm	universal-ner	
from-base	80.37	60.17	78.64	80.61	58.24	77.04	72.51
+KD	81.61	65.94	**80.29**	81.82	59.11	**79.24**	74.67
from-teacher	81.18	62.99	79.50	80.45	58.29	78.20	73.44
+KD	**81.63**	**67.61**	80.18	**82.31**	**61.08**	79.17	**75.33**

Table 3: Performance comparison of XLMR and MDEBERTA models on `from-base` and `from-teacher` initialization. Bold denotes best methods for each data and their average.

Model	Hidden Size	XLM-R			mDeBERTa			Avg
		massive	tsm	universal-ner	massive	tsm	universal-ner	
from-scratch	384	78.05	51.63	36.46	75.93	49.97	44.33	56.09
	768	79.23	54.13	48.29	76.22	49.86	52.13	59.98
from-teacher	384	79.90	58.34	44.87	80.10	54.75	44.87	60.47
	768	81.63	67.61	80.18	82.31	61.08	79.17	75.33

Table 4: Comparative performance of XLM-R and mDeBERTa models across different datasets, initialization methods, and hidden sizes using Knowledge Distillation. `from-scratch` and `from-teacher` use a layer reduction factor of 2.

forms the `from-scratch` method, particularly when trained on the `tsm` and `universal-ner` datasets. While the performance gap is less pronounced for the `massive` dataset, it still favors the `from-teacher` method. These findings suggest that directly copying weights from a larger model can serve as an effective initialization strategy, especially for low-resource scenarios.

Furthermore, the application of knowledge distillation consistently yields performance improvements across all configurations. However, it is noteworthy that the initial weight initialization plays a more critical role in determining overall model performance. The `from-scratch` initialization, even with knowledge distillation, struggles to match the performance levels achieved by the `from-teacher`.

To evaluate the effectiveness of different initialization strategies, we compared models initialized from the teacher to those initialized from the base model. As shown in Table 3, the `from-teacher` approach demonstrates a slight performance advantage over `from-base`, which can be attributed to the teacher model's prior fine-tuning. The application of Knowledge Distillation (KD) improves performance for both initialization methods. For datasets such as `massive` and `universal-ner`, the performance gains are comparable between the two initialization strategies when KD is applied.

Our previous experiments simply reduced the model size by reducing the layer size while retaining the unit size. In practice, however, we might also want to reduce the unit size. In `from-teacher` initialization, the performance in all datasets falls significantly in `tsm` and `universal-ner` by halving the unit size. This performance reduction is expected due to the model's decreased capacity to retain multilingual information. Additionally, the current approach of uniformly copying weights (Xu et al., 2024) may not be optimal for this multilingual distillation task.

4.2 Knowledge of Unseen Languages is Transferrable with Seen Language Teacher Weight Copy

To test the zero-shot cross-lingual performance, we observe two conditions: 1) using the `seen lang` subset as training data for both the student and teacher models, and 2) using the `seen lang` subset as training data for the teacher, then using the English language to fine-tune the student model. The motivation is to observe if the model retains multilingual information from the copy-weight, even when fine-tuned using only English.

Table 5 shows the results of zero-shot cross-lingual generalization. The teacher's accuracy drops significantly compared to the scenario in §4.1. When using the `seen lang` subset for training, we observe similar behavior to the previous results, with a slight difference between in with and without KD in both datasets, unlike the results shown in §4.1. However, without weight-

Model	Initialization	Train Data	XLM-R			mDeBERTa			Avg
			massive	tsm	universal-ner	massive	tsm	universal-ner	
Teacher	from-base	seen-lang	68.22	57.11	84.38	68.54	57.04	87.50	70.47
Student	from-teacher	seen-lang	65.74	54.02	78.47	55.62	47.35	72.77	62.33
	from-scratch	seen-lang	15.10	40.27	14.61	16.36	32.74	14.61	22.28
	from-teacher	english	59.65	53.35	73.73	32.82	46.18	66.61	55.39
	from-scratch	english	7.55	33.24	7.09	3.92	28.89	7.50	14.70

Table 5: Cross-lingual performance (F1-scores %) for XLM-R and mDeBERTa models with different initialization and training data. Knowledge Distillation is used on these models. **Teacher models are trained using `seen-lang`**.

copy, the performance of each data plummets to near-random answers. This shows that weight-copy preserves multilingual knowledge and enables zero-shot cross-lingual generalization in both high and low-resource scenarios.

Using English as the training data deteriorates the performance in `massive`'s without KD setup, yet it gains considerable performance by using KD in the copy-weight initialization. Even when using only the English language, the student still retains the ZSCL generalization performance, albeit with reduced effectiveness, which further strengthens our claim regarding copy-weight multilingual generalization.

On the other hand, `tsm` and `universal-ner` perform similarly to the student model trained on `seen lang`, where using KD yields only a marginal increase. This is attributed to KD needing a sufficient amount of data to be effective.

4.3 Multilingual Distillation is Possible Even if Only English Data is Available

We fine-tune the teacher and student models using only English, as this language is the most widely available. Note that, unlike the experiment done in §4.2, we do not train the teacher model with `seen lang`; we use English instead. We then evaluate it on `unseen lang` to make the results comparable with those in §4.2. We focus on KD since it has shown a consistent pattern in the previous experiments in §4.1 and §4.2.

Table 6 shows the results of the current experiment. Compared to using a fine-tuned teacher model with `seen lang`, we can see that `massive` performance dropped by about 12%, while it slightly improved for `tsm` and `universal-ner`. We argue that `tsm` and `universal-ner` data is non-parallel and contains substantial fewer instances than `massive` . As a result, these performances do not follow the same pattern as `massive`.

Model	Initialization	massive	tsm	universal-ner	Avg
XLMR (T)	from-base	56.42	58.97	77.86	64.42
XLMR (S)	from-teacher	47.85	54.18	69.05	57.03
XLMR (S)	from-scratch	7.50	35.10	9.23	17.28
mDEBERTa (T)	from-base	63.19	59.29	77.93	66.80
mDEBERTa (S)	from-teacher	26.83	44.54	61.97	44.45
mDEBERTa (S)	from-scratch	3.72	28.45	10.12	14.10

Table 6: Cross-lingual performance (F1-scores). (T) **denotes Teacher model trained on English only**; (S) denotes Student model.

Although `from-teacher` exhibited the highest score, there is a significant gap compared to the §4.2 experiment. Having one language trained on the teacher makes copy-weight initialization less effective, yet the model still retains some multilingual capability. In contrast, `from-scratch` performs similarly and near random score[3].

5 Behavior Analysis in Copy-weight Strategy

In §4, we summarized that model initialization strategy significantly impacts transferring multilingual knowledge, with `from-teacher` performing the best. This section provides more detailed analysis related to the model's characteristics when using the copy-weight strategy, such as zero-shot copy classification performance (§5.1), training speed after the weight is copied from the teacher to the student model across different data subsets (§5.2), and 3) performance across different data subsets (§5.3).

The experiments performed in this section will use KD and the setup described in §4.1, with full

[3]Random score is obtained by making all predictions equal to the major class in the dataset.

Training Method	massive	tsm	universal-ner
With Finetune	81.63	67.61	80.18
Without Finetune	38.05	33.57	7.79
Random Score	7.02	33.33	1.75

Table 7: Zero-shot performance by only copying the weight of the respective fine-tuned teacher to their half-layer students. Random score obtained on `universal-ner` is generated through average of random predictions in 30 runs.

hidden size. We focus on analyzing the behavior of the copy-weight strategy.

5.1 Weight Copy model preserve some information even without finetuning

Given the that copy-weight approach is better than the distillation technique itself, we investigate how much multilingual information is retained simply by copying the weights without any additional fine-tuning. Table 7 provides the performance results. We observe that these scores are substantially lower than when fine-tuning is performed. Intriguingly, `massive` and `universal-ner` scores are not as low as random guesses, implying that some knowledge is still retained, though not fully 'connected,' and needs to be fine-tuned. On the other hand, `tsm` shows performance comparable to random guessing. We hypothesize that this is due to the low number of instances in `tsm`, which do not preserve the inherent bias of multilingual knowledge as strongly as the others.

5.2 Weight Copy Models Achieve Higher Performance with Less Data

The experiment in §4.1 demonstrated that the copy-weight approach exhibited higher performance, especially in the `massive` dataset due to its large number of instances. We argue that these results are attributed to better initialization, which enhances data efficiency. To test this hypothesis, we conducted an experiment by creating four subsets of the `massive` dataset, consisting of 1%, 5%, 10%, and 20% of the original data. These subsets were generated using stratified sampling based on the label distribution for each language.

Figure 2 illustrates the results for the three model initialization strategies. We observe a pattern where using more data corresponds to higher scores. The performance order is consistent, with the best scores achieved by `from-teacher` and the worst by `from-scratch`. In the 1% data subset, `from-teacher` achieved around 69% f1-score, showing a significant gap compared to the others, with more than a 20% difference. However, as the dataset size increases, the gap between scenarios becomes smaller, yet `from-teacher` consistently exhibits the best results. This demonstrates that utilizing the teacher's fine-tuned weights, even in a low-resource setting, benefits from the inherited information, providing better scores.

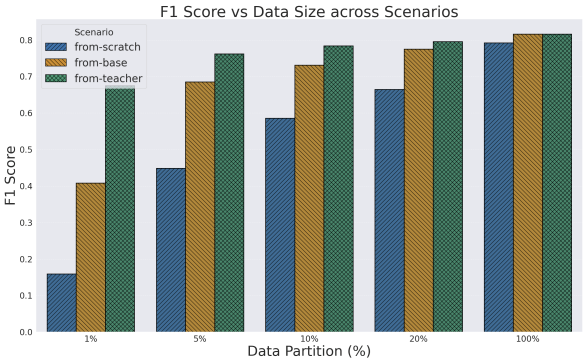

Figure 2: Performance across different data subsets in different initialization strategies.

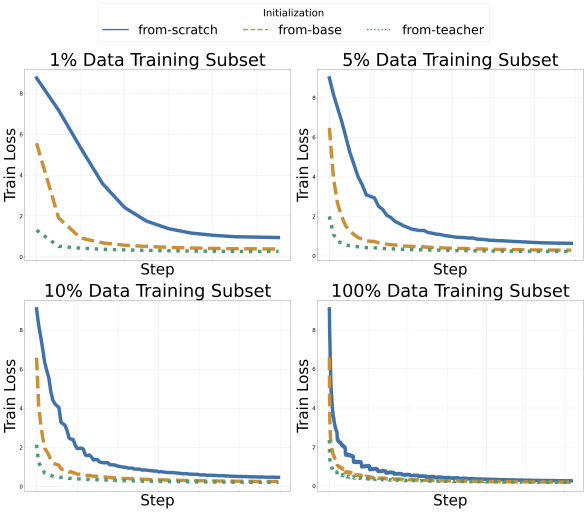

Figure 3: Training loss plot per step across different data subsets.

5.3 Weight Copy Provides Better Initialization – Model Converged Faster

So far, we have explored that different data subsets exhibit different performances across model initialization strategies. This might also correlate with the training speed due to a better start. Thus, we are also interested in exploring learning efficiency by comparing the learning speed of each strategy with each data subset.

The resulting score correlates with the learning

speed, depicted in Figure 3. Using the full data subset, we observe that the order of scenarios sorted by learning speed is similar to the order in Figure 2. With smaller data subsets, the gap in training loss between each model is wide, with `from-teacher` showing the fastest convergence rate. As more data is added, the gap becomes smaller. We posit that this is attributed from the fine-tuned teacher weight-copy, making the model learns faster. Furthermore, this also indicates that copying the teacher's weights in low-resource settings not only improves the score but also accelerates learning speed, reducing the cost of training the model.

6 Related Work

Model Initialization Model initialization is crucial when training a model. Glorot and Bengio (2010) introduced a method to properly initialize the weights of neural networks using a normal distribution to avoid issues related to vanishing and exploding gradients during training. This approach has been extended by several others, such as He et al. (2015), Mishkin and Matas (2016), and Saxe et al. (2014), to add robustness to gradient problems. While these methods address numerical instability, they do not incorporate inherent initial knowledge. Transfer learning (Zhuang et al., 2020; Howard and Ruder, 2018) provides a way to start training a model with better initialization with prior knowledge. We pre-train the model on unlabeled data and then fine-tune it on the desired task. Multilingual models like DeBERTa (He et al., 2021), mBERT (Devlin et al., 2019), and XLM-R (Liu et al., 2019) can be used to train models that handle multiple languages. However, these models require extensive training resources to create.

Knowledge Distillation Knowledge distillation (KD) (Hinton et al., 2015) produces models with fewer parameters (student model) guided by a larger model (teacher model), often resulting in higher quality than models trained from scratch. In NLP, KD can be applied directly to task-specific or downstream tasks (Nityasya et al., 2022; Adhikari et al., 2020; Liu et al., 2020b), or during the pre-training phase of the student model (Sun et al., 2020), which can then be fine-tuned. Several works apply KD during both pre-training and fine-tuning steps (Jiao et al., 2020; Sanh et al., 2019; Liu et al., 2020a). The aspects of the teacher model that the student should mimic can vary; a common approach is for the student to mimic only the probability distribution of the teacher's prediction layer output. However, Jiao et al. (2020) and Sun et al. (2020) also include outputs such as the teacher model's layer outputs, attention layers, and embedding layers. Wang et al. (2020) and Ansell et al. (2023) explore the potential of KD in multilingual settings, with the latter utilizing sparse fine-tuning and a setup similar to Jiao et al. (2020). However, these works do not thoroughly investigate the behavior of the approach, such as the impact of initialization and data size. This research work dives into probing the influence of these components, providing better insight into multilingual performance capability using weight-copy techniques.

7 Conclusion

In this work, we observed the effectiveness of Knowledge Distillation, in multilingual settings, focusing on identifying the factors that significantly influence the performance of student models. Our findings demonstrated that model initialization, specifically through weight copying from a fine-tuned teacher model, plays a crucial role in enhancing the performance and learning speed of student models. This finding was consistent across both high-resource and low-resource datasets, highlighting the importance of weight initialization in retaining multilingual knowledge and facilitating effective KD.

These insights underscore the critical role of initialization in KD, suggesting that simple yet effective strategies, such as weight copying, can lead to substantial performance gains without requiring extensive data or computational resources. This work contributes valuable and practical insights to developing efficient and high-performing multilingual models, particularly in resource-constrained environments. A promising future work is to propose a novel, efficient initialization method that avoids the need for any expensive step, such as pre-training step, in preparing the student model.

8 Limitations

In this work, we focus on sequence classification and token classification tasks, which may not generalize to other tasks, such as Natural Language Generation. The languages observed in this work are those represented in `massive`, `universal-ner` and `tsm`, which do not include every possible language. Additionally, our study is limited to specific model sizes and architectures (e.g., XLM-RoBERTa and

mDEBERTa).

We concentrate on analyzing multilingual capability, specifically zero-shot fine-tuning generalization, rather than zero-shot inference as exhibited in Large Language Models. This focus is due to the amount of computational cost associated with fine-tuning such large models.

Finally, while using unlabeled datasets for distillation may improve the system's performance, it adds another layer of complexity to our work. Analyzing the data for use in a multilingual setting is beyond the scope of this study. We leave this for future work.

Ethical Considerations

This work has no ethical issues, as it focuses on analyzing the inner workings of a multilingual model in knowledge distillation. All artifacts used in this research are permitted for research purposes and align with their intended usage in multilingualism. Additionally, the data utilized does not contain any personally identifiable information or offensive content.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, William L. Hamilton, and Jimmy Lin. 2020. Exploring the limits of simple learners in knowledge distillation for document classification with DocBERT. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 72–77, Online. Association for Computational Linguistics.

Alham Fikri Aji, Genta Indra Winata, Fajri Koto, Samuel Cahyawijaya, Ade Romadhony, Rahmad Mahendra, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Timothy Baldwin, Jey Han Lau, and Sebastian Ruder. 2022. One country, 700+ languages: NLP challenges for underrepresented languages and dialects in Indonesia. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7226–7249, Dublin, Ireland. Association for Computational Linguistics.

Alan Ansell, Edoardo Maria Ponti, Anna Korhonen, and Ivan Vulić. 2023. Distilling efficient language-specific models for cross-lingual transfer. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8147–8165, Toronto, Canada. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266, Marseille, France. European Language Resources Association.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 249–256. JMLR Workshop and Conference Proceedings.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 1026–1034.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020a. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yuang Liu, Wei Zhang, and Jun Wang. 2020b. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106–113.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Stephen Mayhew, Terra Blevins, Shuheng Liu, Marek Šuppa, Hila Gonen, Joseph Marvin Imperial, Börje F. Karlsson, Peiqin Lin, Nikola Ljubešić, LJ Miranda, Barbara Plank, Arij Riab, and Yuval Pinter. Universal ner: A gold-standard multilingual named entity recognition benchmark.

Dmytro Mishkin and Jiri Matas. 2016. All you need is a good init. In *International Conference on Learning Representations (ICLR)*.

Made Nindyatama Nityasya, Haryo Akbarianto Wibowo, Rendi Chevi, Radityo Eko Prasojo, and Alham Fikri Aji. 2022. Which student is best? a comprehensive knowledge distillation exam for task-specific bert models.

Made Nindyatama Nityasya, Haryo Akbarianto Wibowo, Radityo Eko Prasojo, and Alham Fikri Aji. 2021. Costs to consider in adopting nlp for your business.

Sebastian Ruder. 2022. The State of Multilingual AI. `http://ruder.io/state-of-multilingual-ai/`.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In $NeurIPSEMC^2Workshop$.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Andrew M Saxe, James L McClelland, and Surya Ganguli. 2014. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Fei Huang, and Kewei Tu. 2020. Structure-level knowledge distillation for multilingual sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3317–3330, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. 2024. Initializing models with larger ones. In *The Twelfth International Conference on Learning Representations*.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning.

Model	Initialization	#L	#U	massive		tsm		universal-ner	
				NON	KD	NON	KD	NON	KD
XLM-R	from-base (T)	12	768	80.13	–	70.10	–	87.66	–
	from-teacher	6	768	81.18	81.63	62.99	67.61	79.50	80.18
	from-base	6	768	80.37	81.61	60.17	65.94	78.64	80.29
	from-scratch	6	768	75.19	79.23	50.20	54.13	45.68	48.29
	from-teacher	6	384	78.20	79.90	56.74	58.34	47.15	44.87
	from-base	6	384	77.55	79.41	55.03	58.15	45.94	44.24
	from-scratch	6	384	75.27	78.05	50.01	51.63	40.80	36.46
mDeBERTa	from-base (T)	12	768	81.36	–	66.96	–	88.81	–
	from-teacher	6	768	80.45	82.31	58.29	61.08	78.20	79.17
	from-base	6	768	80.61	81.82	58.24	59.11	77.04	79.24
	from-scratch	6	768	75.93	76.22	49.86	51.80	46.21	44.56
	from-teacher	6	384	79.76	80.10	54.59	54,75	62.57	59.72
	from-base	6	384	78.49	78.35	53.94	46.20	62.81	59.24
	from-scratch	6	384	76.22	75.93	49.97	43.39	45.55	44.33

Table 8: F1-scores (%) for XLM-R and mDeBERTa models with different initializations, both with (KD) and without (NON) knowledge distillation, across three datasets. #L: Number of Layers, #U: Number of Units, T: Teacher.

A Additional Experiment Results

A complete and comprehensive experiment results can be seen in Table 8 and Table 9 for all languages training and zero-shot cross-lingual performance in weight-copy, respectively.

B Language Partition Explanation

Detail of language partitions in the training dataset can be seen in Table 12 (`massive`), Table 11 (`tsm`), and Table 10 (`universal-ner`).

Model	Initialization	Training Data	Test Data	massive		tsm		universal-ner	
				NON-KD	KD	NON-KD	KD	NON-KD	KD
XLM-R	from-base (T)	seen lang	unseen lang	68.22	–	57.11	–	84.38	–
	from-teacher	seen lang	unseen lang	64.30	65.74	53.99	54.02	77.05	78.47
	from-base	seen lang	unseen lang	62.77	64.82	52.31	52.36	77.27	79.15
	from-scratch	seen lang	unseen lang	15.08	15.10	37.93	40.27	15.58	14.61
	from-teacher	english	unseen lang	47.23	59.65	54.36	53.35	66.96	73.73
	from-base	english	unseen lang	38.69	46.16	50.47	49.74	64.99	71.73
	from-scratch	english	unseen lang	5.25	7.55	34.39	33.24	9.21	7.09
mDeBERTa	from-base (T)	seen lang	unseen lang	68.54	–	57.04	–	87.50	–
	from-teacher	seen lang	unseen lang	50.91	55.62	48.10	47.35	73.28	72.77
	from-base	seen lang	unseen lang	47.59	54.90	46.66	49.04	69.13	73.56
	from-scratch	seen lang	unseen lang	14.80	16.36	39.59	32.74	15.58	14.61
	from-teacher	english	unseen lang	20.98	32.82	43.14	46.18	59.10	66.61
	from-base	english	unseen lang	16.23	33.64	43.38	48.32	53.37	65.34
	from-scratch	english	unseen lang	7.09	3.92	38.45	28.89	8.26	7.50

Table 9: F1-scores (%) for zero-shot cross-lingual generalization in Knowledge Distillation for XLM-R and mDeBERTa models. We fine-tune the teacher model using `seen lang` and fine-tune the student model according to the training data provided. NON-KD follows the student model configuration. All models contain 6 layers, except for the teacher models which have 12 layers.

Code	Language	Code	Language	Code	Language
ceb	Cebuano	en	English	sk	Slovak
da	Danish	hr	Croatian	sr	Serbian
de	German	pt	Portuguese	sv	Swedish
ru	Russian	tl	Tagalog	zh	Chinese

Table 10: Languages in the `universal-ner` dataset

Code	Language	Code	Language	Code	Language
arabic	Arabic	german	German	portuguese	Portuguese
english	English	hindi	Hindi	spanish	Spanish
french	French	italian	Italian		

Table 11: Languages in the `tsm` dataset

Code	Language	Code	Language	Code	Language
af-ZA	Afrikaans	it-IT	Italian	ru-RU	Russian
am-ET	Amharic	ja-JP	Japanese	sl-SL	Slovanian
ar-SA	Arabic	jv-ID	Javanese	sq-AL	Albanian
az-AZ	Azeri	ka-GE	Georgian	sv-SE	Swedish
bn-BD	Bengali	km-KH	Khmer	sw-KE	Swahili
ca-ES	Catalan	ko-KR	Korean	hi-IN	Hindi
zh-CN	Chinese (China)	lv-LV	Latvian	kn-IN	Kannada
zh-TW	Chinese (Taiwan)	mn-MN	Mongolian	ml-IN	Malayalam
da-DK	Danish	ms-MY	Malay	ta-IN	Tamil
de-DE	German	my-MM	Burmese	te-IN	Telugu
el-GR	Greek	nb-NO	Norwegian	th-TH	Thai
en-US	English	nl-NL	Dutch	tl-PH	Tagalog
es-ES	Spanish	pl-PL	Polish	tr-TR	Turkish
fa-IR	Farsi	pt-PT	Portuguese	ur-PK	Urdu
fi-FI	Finnish	ro-RO	Romanian	vi-VN	Vietnamese
fr-FR	French	he-IL	Hebrew	cy-GB	Welsh
hu-HU	Hungarian	hy-AM	Armenian	id-ID	Indonesian
is-IS	Icelandic				

Table 12: Language codes and corresponding language names in `massive` dataset.