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 a smaller model 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 finding shows 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

Utilizing pre-trained multilingual models is often useful when the number of available data is limited. These models merit better initialization, i.e., initial multilingual knowledge, which results in better learning convergence and better performance than training from scratch (Raffel et al., 2020; Devlin et al., 2019; Liu et al., 2019). these models typically have a large number of parameters. This becomes crucial if we aim to produce low-cost, low-resourced language models (Nityasya et al., 2021).

To reduce the cost, recent work proposes utilizing Knowledge Distillation (KD) in multilingual setting (Ansell et al., 2023), which follows the training approach of TinyBERT (Jiao et al., 2020), involving a two-step distillation process where the student model is pre-trained before fine-tuning to acquire multilingual knowledge. This approach, however, requires substantial data for the pre-training step, thus increasing computational costs. Additionally, it remains unclear which components of the distillation process have the most significant impact on performance.

We address this research gap by analyzing the impact of KD and model initialization on the performance of multilingual models. In this work, we opt for an efficient yet effective initialization method by copying the weights from the teacher model to the student model in an alternate manner, also done by DistilBERT (Sanh et al., 2020). Although this approach has been previously used, its effectiveness in multilingual scenarios has not been fully explored. Therefore, we aim to analyze the impact of KD and weight copying on multilingual model performance.

Our contributions to this work are as follows:

1. We found 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. Our experiments revealed that the weight-copy approach leads to faster learning speeds and exhibits multilingual knowledge, even without fine-tuning. Among the two weight-copying strategies we compared, directly copying the teacher's weights resulted in the fastest learning speed across different data sizes.

2 Background

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

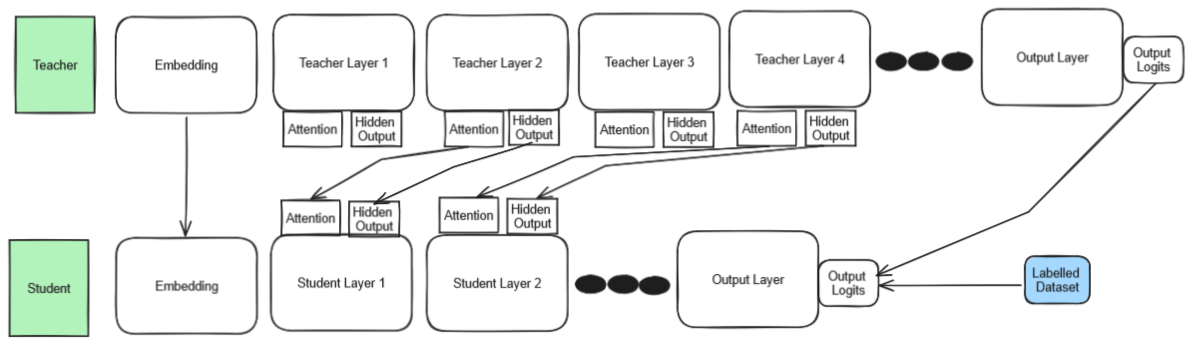

Figure 1: Overall architecture of Knowledge Distillation used in this paper where the teacher distills its knowledge using Mean Square Loss (MSE). The student model also learns from the labeled dataset.

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). While this method has proven effective, it has not been thoroughly explored in multilingual settings, which presents an intriguing area to observe.

The methodology of these components used in this work is elaborated in §2.1 (Distillation Architecture) and §2.2 (Model Initialization).

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. l and h denote the indices of the model layers and attention heads. 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]. We denote the k-th hidden layer of the teacher and the corresponding i-th hidden layer of the student, where the mapping formula, based on the best ablation results of Jiao et al., 2020, is defined as follows:

$$k = i \cdot divisor \quad \text{for} \quad i, k \in \mathbb{Z}^+ \quad (5)$$

We denote $divisor$ as the division number of the teacher's self-attention layers relative to the student's.

In the original formulation, $\mathcal{L}_{pred}$ has a temperature hyperparameter that we can control to set the smoothness of the output distribution. However, this hyperparameter can be ignored according to (Jiao et al., 2020).

The KD loss can be formulated as follows:

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

Information	massive	tsm
Number of Training Data	11,514	1,839
Number of Class	60	3
Number of Language	52	8
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"

Table 1: Data statistic in massive and tsm. Each language consists of the same number of instances in both datasets. unseen lang denotes language subset used in the zero-shot cross-lingual experiment. The rest of the languages are categorized as seen lang

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

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:

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

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 Multilingual Transferability in KD

The ability of knowledge distillation (KD) to transfer knowledge across multiple languages efficiently remains unexplored. These intriguing issues will be elucidated with detailed analysis in the respective subsections:

1. 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 (see §3.2).

2. 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 (see §3.3).

3. Building a multilingual dataset is tedious, thus leading researchers to opt for using one

Model Initialization	Num Layer	Num Units	massive		tsm	
			NON-KD	KD	NON-KD	KD
from-base (Teacher)	12	768	80.13%	-	70.10%	-
from-teacher	6	768	81.18%	**81.63%**	62.99%	**67.61%**
from-base	6	768	80.37%	81.61%	60.17%	65.94%
from-scratch	6	768	75.19%	79.23%	50.20%	54.13%
from-teacher	6	384	78.20%	**79.90%**	56.74%	**58.34%**
from-base	6	384	77.55%	79.41%	55.03%	58.15%
from-scratch	6	384	75.27%	78.05%	50.01%	51.63%

Table 2: F1-scores of the experiment using three different model initializations, both with (KD) and without knowledge distillation (NON-KD).

language. It is desirable if this single language can achieve cross-lingual generalization (Artetxe and Schwenk, 2019). Can KD perform distillation effectively using only one language? (see §3.4)

To analyze these issues, we provide the experimental setup used in §3.1.

3.1 Setup

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

Data In the following experiments, our work focuses on multilingual classification tasks. We utilized massive (FitzGerald et al., 2022) and Tweet Sentiment Multilingual dataset (denoted as tsm) (Barbieri et al., 2022). 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). These data comprise 52 and 8 languages, respectively. These languages are then 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.

Model We used the transformers library (Wolf et al., 2020) and the off-the-shelf implementation of the 'xlm-roberta-base' model (Liu et al., 2019) for this work. 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' (from-base), copying from the fine-tuned teacher (from-teacher), and initializing without copying from any model (from-scratch). We hypothesize that the from-teacher strategy, containing more information, should perform the best. However, we compared their performances to understand the differences between these strategies. The copy-weight methods follow §2.2.

Training Setup 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 §3.2 and §3.3, and 100 steps for §3.4. For tsm, we set the evaluation steps to 500, 250, and 60 for §3.2, §3.3, and §3.4, 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. The depicted scores in the following results are the averages of these runs, except for the teacher model, where we selected one of the best models for the distillation process.

3.2 Weight Copy Transfers more information vs Distillation Loss

We compare the training performance with and without knowledge distillation to support this finding, denoted as KD and NON-KD, respectively. To run the KD, The teacher model is trained on all available languages in the respective datasets in the training set and then evaluated on the performance of the corresponding test set.

Model Initialization	Training Data	Test Data	massive		tsm	
			NON-KD	KD	NON-KD	KD
from-base (Teacher)	seen lang	unseen lang	68.22%	-	57.11%	-
from-teacher	seen lang	unseen lang	64.30%	**65.74%**	53.99%	**54.02%**
from-base	seen lang	unseen lang	62.77%	64.82%	52.31%	52.36%
from-scratch	seen lang	unseen lang	15.08%	15.10%	37.93%	40.27%
from-teacher	english	unseen lang	47.23%	**59.65%**	**54.36%**	53.35%
from-base	english	unseen lang	38.69%	46.16%	50.47%	49.74%
from-scratch	english	unseen lang	5.25%	7.55%	34.39%	33.24%

Table 3: F1-Score for zero-shot cross-lingual generalization in Knowledge Distillation. We fine-tune the teacher model using seen lang and fine-tune the student model according to the training data provided in this table. NON-KD follows the student model configuration. The models used in this table contain 6 number of layers.

Table 2 provides the performance of such a setup. It can be seen that student models that copy weights from these teachers outperform their teachers on the massive dataset in both NON-KD and KD performance, achieving more than 1.5 F1-score points higher than the teacher in some cases. These models benefit from the abundant data available in massive. Conversely, experiments using the tsm dataset do not exhibit similar improvements.

Regarding model initialization strategy, the from-teacher copy-weight strategy consistently performs the best, followed by from-base and from-scratch in both datasets. This corresponds to the level of inherent task knowledge in each setting. from-base yields a small decrease compared to from-teacher. Without copy-weight, the performance gap is moderate in massive but significantly worse in tsm. This result underlines the **impact of knowledge preservation of copy-weight on performance.**

Using knowledge distillation consistently improves performance for both datasets, with the tsm dataset model initialization seeing a significant performance increase of about 3-5%. It is evident that KD helps to increase performance further, especially when using low-resource data. However, when comparing KD to model initialization settings, it is clear that **having better initialization is more important than utilizing KD, especially for low-resource data**. For instance, tsm has the worst performance with from-scratch model initialization, and even using KD with this initialization cannot surpass from-base's score.

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. By halving the unit size, the performance in both datasets falls, which is expected due to the lower capacity of the model to save multilingual information. However, these results show a similar pattern to the full number of hidden units. Using KD marginally increases performance, and initialization using copy-weight boosts the model significantly. Note that from-scratch has comparable performance when having a full and half number of units. In contrast, halving the hidden units of a model initialized by copying from another model shows a significant gap, indicating that **the number of units matters in preserving the information of weight-copy.**

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

Based on the results in §3.2, we see a similar pattern exhibited by training a copy-weight model with half the hidden units and the full number of hidden units. Thus, we can focus on using the full hidden units in subsequent methods to save experiment time. 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 3 shows the results of zero-shot cross-lingual generalization. The teacher's accuracy drops significantly compared to the scenario in §3.2. When using the seen lang subset for training, we observe similar behavior to the previous results, with a slight difference between KD and NON-KD in both datasets, unlike the results shown in §3.2.

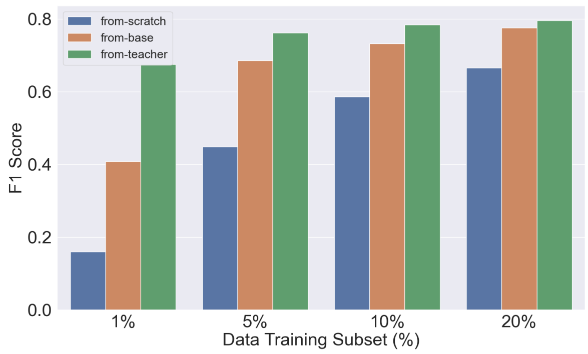

Figure 2: Performance across different data subsets in different initialization strategies

Model Initialization	massive	tsm
from-base (Teacher)	56.42%	58.97%
from-teacher	47.85%	54.18%
from-base	42.05%	51.56%
from-scratch	7.50%	35.10%

Table 4: F1-Score of student model performance fine-tuned with English dataset using knowledge distillation. The teacher is also trained in English, not seen lang, shown in Table 2. The performance is then evaluated to unseen lang

However, without weight-copy, the performance plummets to near-random answers (around 7% and 33%). 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 NON-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 performs 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.

3.4 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 §3.3, 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 §3.3. We focus on KD since it has shown a consistent pattern in the previous experiments in §3.2 and §3.3.

Table 4 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. We argue that tsm's data is non-parallel and contains few instances (1,839). As a result, the performance of tsm does not follow the same pattern as massive. This increase in tsm results is due to better teachers than in the previous experiment.

It can be seen that the performance of massive degrades by about 12% for from-teacher and 4% for from-base compared to using a fine-tuned teacher with seen lang. What is interesting is that although from-teacher exhibited the highest score, there is a significant gap compared to the §3.3 experiment using the seen lang fine-tuned teacher model. **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 the random score[2].

4 Behavior Analysis in Copy-weight Strategy

In §3, 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: 1) zero-shot copy classification performance (§4.1), 2) training speed after the weight is copied from the teacher to the student model across different data subsets (§4.2), and 3) performance across different data subsets (§4.3).

The experiments performed in this section will use KD and the setup described in §3.2, with full hidden size. We focus on analyzing the behavior of the copy-weight strategy.

Training Method	massive	tsm
With Finetune	81.63%	67.61%
Without Finetune	38.05%	33.57%
Random Score	7%	33.33%

Table 5: Zero-shot performance by only copying the weight of the respective fine-tuned teacher to their half-layer students.

4.1 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 5 provides the performance results. We observe that these scores are substantially lower than when fine-tuning is performed. Intriguingly, `massive` 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 `massive`.

4.2 Weight Copy model require less data to Converge

The experiment in §3.2 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.**

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

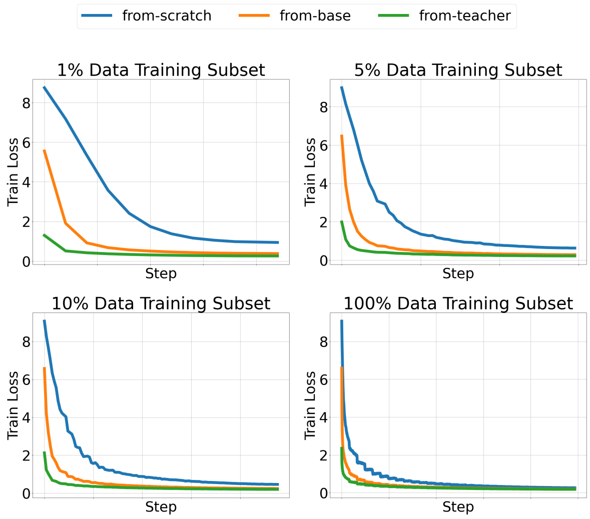

Figure 3: Training loss plot per step across different data subsets.

4.3 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. This 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.**

5 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.

6 Conclusion

In this work, we observed the effectiveness of Knowledge Distillation (KD), in multilingual settings, focusing on identifying the factors that significantly influence the performance of student models. Our finding 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 in preparing the student model.

7 Limitations

In this work, we focus solely on the classification task, which may not generalize to other tasks, such as Natural Language Generation. The languages observed in this work are those represented in `massive` and `tsm`, which do not include every possible language. Additionally, our study focuses on a particular model size and architecture (e.g., XLM-RoBERTa). Different models or architectures might exhibit different behaviors under similar conditions, so the findings may not generalize to all multilingual models. The experiments were conducted with a fixed set of hyperparameters. 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.

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.

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*.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

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

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.