electronics

MDPI

Article

Distilling Monolingual Models from Large Multilingual Transformers

Pranaydeep Singh *, Orphée De Clercq and Els Lefever

LT3, Language and Translation Technology Team, Department of Translation, Interpreting and Communication, Ghent University Groot-Brittanniëlaan 45, 9000 Ghent, Belgium

* Correspondence: pranaydeep.singh@ugent.be

Abstract: Although language modeling has been trending upwards steadily, models available for low-resourced languages are limited to large multilingual models such as mBERT and XLM-RoBERTa, which come with significant overheads for deployment vis-à-vis their model size, inference speeds, etc. We attempt to tackle this problem by proposing a novel methodology to apply knowledge distillation techniques to filter language-specific information from a large multilingual model into a small, fast monolingual model that can often outperform the teacher model. We demonstrate the viability of this methodology on two downstream tasks each for six languages. We further dive into the possible modifications to the basic setup for low-resourced languages by exploring ideas to tune the final vocabulary of the distilled models. Lastly, we perform a detailed ablation study to understand the different components of the setup better and find out what works best for the two under-resourced languages, Swahili and Slovene.

Keywords: knowledge distillation; low-resource NLP; sustainable NLP; language modeling

Citation: Singh, P.; De Clercq, O.; Lefever, E. Distilling Monolingual Models from Large Multilingual Transformers. *Electronics* **2023**, *12*, 1022. https://doi.org/10.3390/electronics12041022

Academic Editor: Ruifeng Xu

Received: 14 January 2023
Revised: 14 February 2023
Accepted: 16 February 2023
Published: 18 February 2023

1. Introduction

The advent of extremely large language models (LLMs) in the past decade has pushed Natural Language Processing (NLP) for under-resourced languages beyond all foreseen expectations, while the building and training of these LLMs has been an impetus for low-resource NLP, the deployability and sustainability of these technologies for real-world use cases is an often ignored secondary aspect. Even though multilingual models such as mBERT [1] and XLM-R [2] excel at low-resource and multilingual NLP, they often fail when it comes to this second aspect because they are extremely large language models with vocabularies of hundreds of languages, which may not be necessary for the deployment of a model for a single low-resourced language. Unlike for high-resourced languages, under-resourced languages often lack the availability of a single monolingual language model, such as CamemBERT [3] for French or RobBERT [4] for Dutch, thus making large jointly trained multilingual models a necessary evil; while one can argue that mBERT and XLM are still deployment-friendly in some ways, the trends toward an exponential rise in parameters will soon make it impossible to deploy research-grade released models. For example, this occurs in the mT5-XXL (13 billion parameters) [5] and the Turing ULR (4.6 billion parameters) [6] series of models, which are currently state of the art on the XTREME [7] data set—a comprehensive benchmark for cross-lingual transfer learning for a large variety of NLP tasks and languages.

While there have been significant strides forward in reducing model footprints, inference, and training times with methodologies such as Distillation, Quantization, and Pruning, these methodologies are often tested in a general direction, i.e., reducing a multilingual model as a whole, or in a task-specific setting, i.e., creating a smaller model specialized for a particular task. In this work, we attempt to explore the consequences of using the ideas behind knowledge distillation and applying these to large pre-trained multilingual

models, to filter knowledge specific to a target language into a new, smaller, and faster student language model which performs identically to or even outperforms the teacher in some cases. The main contribution of this paper is to dive deep into standard knowledge distillation practices and explore optimal strategies to distill individual target languages from a large multilingual model.

The first objective of the proposed research is to explore the standard knowledge distillation setup designed for generic full-model distillation for two widely used multilingual models, i.e., multilingual-BERT (mBERT) and XLM-RoBERTa (XLM-R). Important to note is that we attempt to only keep information for a single target language for the student. We build upon the pilot experiments for *Eliquare*, first proposed in Singh and Lefever [8], and perform all experiments on a set of six carefully selected languages accounting for as much variation as possible with regard to their typologies, language families, and available resources. We consider Dutch and French to be representative of high-resourced languages, Hindi and Hebrew are considered moderately resourced languages, and Swahili and Slovene are representatives of low-resourced languages. For each language, we evaluate the obtained distilled students on a set of two downstream tasks: one being a syntactic word-level task such as Part-of-Speech Tagging and the other a semantic sentence-level task such as Sentiment Analysis.

A second, and perhaps more vital objective of this research is to propose ideas that specifically benefit the construction of students for low-resourced languages, i.e., Swahili and Slovene in our case. We attempt to do this in two stages. Firstly, we explore the principles behind altering the vocabularies of the final student to better suit the low-resource setting. While joint models have large combined vocabularies which assist in multilingual aspects, for a distilled student model only the vocabulary of a single target language is required. While the high-resourced languages used in our work (Dutch and French) have enough sub-words in the multilingual vocabulary to adequately represent the language space, the middle- and under-resourced languages have an extremely poor representation. In mBERT, for example, a medium-resourced language such as Hebrew has around 2483 sub-words in the vocabulary accounting for approximately 2% of the whole vocabulary, while Thai only has 370 sub-words, amounting to around 0.3% of the vocabulary. We, therefore, explore techniques to reduce the vocabulary sizes both pre- and post-distillation while keeping the performance consistent across all benchmarks. Secondly, we perform a detailed ablation study to explore what components and hyper-parameters specifically impact the performance of the distilled student in the low-resource setting. We specifically dive deeper into the two most vital components of the distillation framework: the loss and softmax temperature.

The remainder of this paper is structured as follows. In Section 2, we first describe relevant related research on knowledge distillation and shortcomings of large multilingual language models for low-resourced settings, and Section 3 discusses the fundamental principles of classic knowledge distillation and builds from the DistilBERT [9] setup towards a language-specific distillation setting. Section 4 discusses the experimental setup and results of the basic setups and demonstrates the viability of the proposed *Eliquare* methodology. Sections 5 and 6 further venture into advanced modifications possible to the distillation setup, to suit a low-resource language setting, while Section 5 discusses the concept of altering the vocabularies of the multilingual models, to only accommodate a low-resourced language, while also speeding up the distillation process further. Section 6 discusses the impact of some of the key hyper-parameters and their impact on the student models for Slovene and Swahili. Section 7 concludes this paper by summarizing our findings and suggesting ideas for future research.

2. Related Work

This work takes inspiration from two research strands: the methodology of language-specific distillation stems from research into sustainability and efficiency in NLP, while the application for low-resourced settings stems from work about the deficiencies of large

multilingual models. Therefore, the related work is divided into two sections. Section 2.1 covers the sustainability for large language models research strand and more specifically zooms in on different methodologies that have been employed for knowledge distillation through the last decade, as well as their evolution. Section 2.2 covers work delving into the analysis of multilingual LLMs to decipher our understanding of the inner workings of multilingual representations, and to some extent, the failures of models such as mBERT and XLM-R.

2.1. Knowledge Distillation

Research into the reduction in deep learning model sizes, inference speeds, training times, and therefore, distillation, is almost as old as deep learning itself. The term distillation is also used in the context of data distillation [10], which is an entirely independent strand of research that focuses on generating high-fidelity data summaries for large datasets. In this work, however, we focus solely on model distillation, more commonly referred to as knowledge distillation. While knowledge distillation as we know it today was initially popularized by Hinton et al. [11], the initial concepts of the distillation methodology were formalized by Bucilua et al. [12] for distilling the prediction power of an ensemble of teacher models into a single student. The basic fundamentals behind distillation as defined by Bucilua et al. have remained consistent over the years. Given a teacher function $f(t)$, learn a student function $f(s)$ such that $f(s)$ is an approximation of $f(t)$; while this earlier work explored generating new pseudo-training samples from the approximate distribution of the training data to learn $f(s)$, Hinton et al. introduced the more reliable methodology of imitating the prediction by $f(t)$ directly penalizing the distance in logits between $f(s)$ and $f(t)$. They demonstrated the viability of this approach in various settings, for example, in Image Recognition and Automatic Speech Recognition.

Initial work in knowledge distillation (KD) primarily focused on ensemble models and applications such as Automatic Speech Recognition [13] and Autonomous Driving [14]. In NLP, work has been performed with KD for specific task-based settings such as Question Answering [15], Machine Translation [16], Intent Classification for Voice Assistants [17], and Text Generation [18]. One of the seminal works, by Tang et al. [19], attempted to distill task-specific knowledge for sentence-pair tasks such as Natural Language Inferencing (NLI) and Paraphrasing into a pair of Siamese single-layer BiLSTMs and showed comparable performance to BERT with only one-hundredth of the parameters. They also opted to use Mean-Squared Error to align the student and the teacher and found it to work better in this setting than the standard cross-entropy with soft targets that was proposed by Hinton et al. [11].

With the construction of Large Language Models (LLMs) such as BERT [1] and XLM-RoBERTa [2] and their wide adaptation for a wide range of NLP tasks by fine-tuning them, there has emerged a need for general model reduction strategies to be able to have miniaturized transformers that could be deployed in practice on, for example, mobile devices and other low-compute settings. A number of distillation setups for BERT have been proposed, such as DistilBERT [9], BERT-PKD [20], TinyBERT [21], and MobileBERT [20]. In what follows, we will outline the differences between these distillation methods and compare their performance on the well-known GLUE (General Language Understanding Evaluation) benchmark [22]. GLUE is a set of nine carefully selected and varied tasks that cover a variety of text domains, difficulty, and dataset sizes in the English language and is often used as a staple benchmark to evaluate language understanding of large models in English. A summary of the discussed distillation methodologies that we will build upon is provided in Table 1.

Table 1. An overview of key distillation strategies. Column 2 describes the student model, column 3 the distillation loss, columns 4, 5, and 6 the inference speed, model size, and performance on benchmarks relative to the teacher (BERT-base-uncased), respectively. Wherever possible, the speeds, sizes, and performances refer to the 6-layer variant to allow comparison with other methodologies.

Methodology	Student	Objective	Speed	Size	Performance
DistilBERT	6-layer 768 hidden size	KD Loss + Cosine	2.0×	1.6×	0.966×
BERT-PKD	6,4-layer 768 hidden size	Layer-wise Minimization	2.0×	1.6×	0.975×
TinyBERT	6,4-layer 768 hidden size	Layer-wise + Attention-heads Minimization	2.0×	1.6×	0.998×
MobileBERT	12-layer 128 hidden size	KD Loss + Feature Map + Attention Loss	5.5×	4.3×	0.992×

DistilBERT, the seminal work by Sanh et al. [9], largely builds on the basic distillation setup from Hinton et al. [11] by distilling a six-layer BERT student from BERT-base, with a triplet loss consisting of cross-entropy and Cosine Embedding between the student and teacher soft targets and the standard Masked Language Model (MLM) Loss. The resulting student, while having 40% lesser parameters and 60% faster inference speed compared to the teacher, retains 97% of the performance of BERT-base on the GLUE benchmark dataset.

BERT-PKD (Patient Knowledge Distillation) [20] first introduced the methodology of using the middle layers in the teacher transformer as additional signals to assist the student's learning. They introduce additional loss components which assist each layer in the student by learning from the output of an analogous layer in the teacher. This patient layer-by-layer distillation is only performed for each training sample's (CLS) token rather than all the sub-words in a sample to speed up the learning process. The experiments demonstrate that the patiently distilled students outperform basic distilled students without the additional loss components. However, the resulting BERT-PKD with six layers and identical parameter counts is not able to consistently outperform DistilBERT on the GLUE benchmark, with BERT-PKD performing better on three out of six tasks tested and DistilBERT performing better on the remaining three, despite the more fine-grained loss components introduced in BERT-PKD. A potential explanation could be the better initialization strategy opted for DistilBERT, i.e., using alternate layers in the teacher transformer to directly initialize the hidden layers of the student.

TinyBERT [21] attempts to perform task-specific distillation in a two-step process, by first distilling a smaller, four-layer masked language model from BERT-base, and second, by fine-tuning for task-specific learning stages with augmented data from the original task dataset. The MLM distillation goes beyond the standard KD setup and applies a layer-by-layer distillation methodology, where the loss is defined differently for the embedding layer, the hidden layers, and the prediction layer. This allows the loss of the hidden layers to include additional attention components from each head aside from the standard hidden-state components, allowing for significantly fine-grained distillation. This advanced setup helps TinyBERT retain more than 96.8% of the teacher's performance on the GLUE benchmark. Further, the ablation studies clearly demonstrate that attention-specific distillation losses have a big impact on the distillation performance, even more so than the hidden state loss components.

MobileBERT [20] follows the completely novel direction of reducing the width (hidden size) of a transformer instead of the depth (layers). Unlike the other described distillation strategies, the teacher is retrained with inverted bottleneck layers which reduce the feature-map sizes of the original BERT-large model. The work further introduces additional strategies to improve the student, with loss components to transfer information from the

feature maps and attention heads, replacing the layer normalization with a basic linear transformation and factorizing the embedding layer using 1D convolutions. The student model is just as big as a BERT-base model in terms of layers, but with a hidden size of 128 compared to BERT-base's 768, it is 4.3 times smaller and 5.5 times faster for inference than BERT-base, while being only 0.6% worse than BERT-base on the GLUE benchmark. As such, it outperforms the previously described approaches. This approach, however, has substantial computation requirements because it is required to retrain the teacher BERT-Large.

2.2. Multilinguality in LLMs

Models such as mBERT and XLM introduced the idea of training large multilingual models with multitudes of zero-shot cross-lingual transfer tasks such as unsupervised NMT (Neural Machine Translation) [23] and word alignment without parallel data [24]. Conneau et al. [25] further demonstrated that cross-lingual transfer performs adequately in highly unlikely scenarios, for example, when the training and test data are from two different domains, or even when languages do not have any shared vocabulary. Despite the many applications of multilingual modeling, there was until recently little understanding of how the different representations for each language were structured inside a multilingual LLM. Pires et al. [26] initially began exploring the limits of mBERT's multilingual representations with a set of probing experiments. They discovered that while mBERT does seemingly learn multilingual representations with no cross-lingual signals during training, these representations are highly limited. Moreover, they found that cross-lingual transfer tends to work best with typologically similar languages with high lexical overlap. Further probing also demonstrated mBERT's surprising flexibility towards code-switching and changes in scripts. Moreover, it was discovered that representations of the same sentence in two different languages tend to be very similar, especially in the earlier layers. The authors further hypothesize that mBERT and similar multilingual models' ability to create a shared space for multiple languages comes from being forced to align URLs and numbers, which forces other word pieces to be aligned closely according to the distributional hypothesis, which states that linguistic items having a similar distribution over a corpus are likely to have similar meanings.

Wu and Dredze (2020) [27] performed a seminal study criticizing mBERT's representation of low-resourced languages. They found that while mBERT performs well above baseline scores for the high/medium-resourced languages based on Wikipedia sizes, the performance for the bottom 30% languages (including languages such as Mongolian and Yoruba with less than 1 million sentences on Wikipedia) is below baseline; the results demonstrate that there is a direct correlation between the Wikipedia size of a language and its resulting performance for downstream tasks in that same language, but this does not seem to be the only factor. Other variables such as the size of the downstream task dataset and representation in the multilingual vocabulary seem to be pivotal as well. The authors also directly compared mBERT to monolingual models for low-resourced languages such as *Latvian*, *Yoruba*, *Mongolian*, and *Afrikaans* and concluded that despite mBERT's bias to high-resourced languages, the multilingual objective still benefits low-resourced languages as the multilingual model outperforms the monolingual models in these languages. This finding once more emphasizes that multilingual models are not only useful for cross-lingual transfer tasks but that their monolingual capabilities for low-resourced languages can have a high impact.

In this work, we attempt to connect these two research directions, viz. multilingualism and knowledge distillation, to solve a fundamental obstruction in the global adaptation of large multilingual models. We apply the techniques discussed in Section 2.1 to construct student models that are able to replicate the monolingual performance of the teacher with a significant size reduction and inference speed increase.

3. Language Distillation Setup

We begin the system description by explaining the fundamental principles behind a distillation setup in more detail. While there has been work that is an exception that forgoes the standard logit setup and uses ideas such as mutual information [28] and graph-based methods [29], most distillation methodologies work with a few common principles at the core. Distillation, as previously described, can be simply thought of as the task of finding the approximation

$$f_s(x) \approx f_t(x) \tag{1}$$

where $f_s(x)$ is the student model's final output for the training data x, and $f_t(x)$ is a teacher model's final output on the same data. There can be three broad variables in a distillation setup. Firstly, the data used for distillation, which determines the type of knowledge being distilled. For instance, to distill a specialized model for Natural Language Inference (NLI) (NLI is a sentence-pair task that given a premise, evaluates if a hypothesis is an entailment, a contradiction, or unrelated to the premise), only information vital for NLI needs to be filtered from the teacher, and this can be conducted by imitating the teacher's knowledge for an NLI dataset, therefore implying that any stored information not relevant to NLI can be forgotten. A few approaches experiment with augmenting data to boost the learning towards a target task, but this is usually useful in task-specific settings where labeled data is a requirement for distillation. The second variable can be the loss function. The loss function essentially determines how we choose to compare the student to the teacher during the learning stage. Given a loss metric $L(x,y)$ and a teacher and student prediction on a sample i represented by $f_t(i)$ and $f_s(i)$, a minimization objective over a dataset of size N can be defined as

$$min. \sum_{i=0}^{N} L(f_s(i), f_t(i)) \tag{2}$$

The third and final variable can be how the student model is set up, primarily, the architecture and the initialization. While most approaches work with an architecture identical to the teacher but with a smaller number of layers, there has been work that adopts simpler architectures for the student than for the teacher. A number of initialization strategies have also been explored since a better initialization can heavily impact the distillation outcome, as shown by Turc et al. [30].

Regarding the first variable, i.e., the distillation data, our goal is to distill knowledge relevant to a single target language which is why we use the entire latest Wikipedia dump for the target language. The minimization objective for that given target language t can then be simply modified as

$$min. \sum_{i:i \in N_t}^{N} L(f_s(i), f_t(i)) \tag{3}$$

For the second variable, i.e., the distillation objective, Hinton et al. [11] introduced two vital contributions, which have become fundamental building blocks of most distillation setups since. Firstly, the error function $L(x,y)$ is defined as the cross-entropy between the student and teacher logits:

$$L_{CE}(f_s(i), f_t(i)) = f_t(i) \times log(f_s(i)) \tag{4}$$

Secondly, Hinton et al. also introduced the concept of softmax temperature. Instead of using logits from the teacher directly in the error function L_{CE}, they propose using soft targets instead, determined by a preset temperature value. Given a temperature value τ and $f(x)_k$ representing the k^{th} output logit given K classes, the soft targets can be generated with

$$g(x) = \sum_{k=0}^{K} \frac{exp(f(x)_k)/\tau}{\sum_j exp(f(x)_k)/\tau} \tag{5}$$

Thus, softening the probability distribution of the logits if $\tau > 1$ or hardening the distribution if $\tau < 1$. Softening the targets can produce stable training that reduces the impact of noisy labels from the teacher model, while hardening can be more useful for faster convergence when distillation data is hard to come by. Sanh et al's [9] setup inherits from the temperature-based soft targets and uses cross-entropy between the soft targets as the error function $L(x,y)$. Additional loss functions L_{cosine} and a standard L_{mlm} for Masked Language Modeling defined below, are used in addition to L_{CE}

$$L_{cosine}(f_s(i), f_t(i)) = \sum_{i \epsilon L} 1 - cos(f_t(i), f_s(i)) \tag{6}$$

$$L_{mlm}(f_s(i), y(i)) = y(i) * log(f_s(i)) \tag{7}$$

While L_{cosine} is expected to minimize the cosine distance between the soft targets and the student logits, L_{mlm} adds an additional component that learns directly from the data $y(i)$ instead of the teacher outputs. This can serve as a self-correction for those examples where the teacher is not always reliable, while also speeding up training by adding an additional learning signal directly from the ground truth. The three losses are combined with a preset weighted sum,

$$L = \alpha_{CE} L_{CE} + \alpha_{cosine} L_{cosine} + \alpha_{mlm} L_{mlm} \tag{8}$$

While for the initial setups, we inherit the preset weights ($\alpha_{CE} = 5, \alpha_{cosine} = 1$, $\alpha_{mlm} = 2$) and softmax temperature ($\tau = 2.0$) from Sanh et al. [9], we discuss the impact of these components further in Section 6.

For the third and final variable, i.e., the student model's setup, we use an architecture identical to the teacher, but with 6 encoder layers, in contrast to the 12 teacher layers. We attempt two alternate setups by changing the vocabulary of the teacher pre-distillation or of the student directly through post-distillation. This is further covered in Section 5, as the initial experiments did not involve any changes to the vocabulary. Another important part of this variable is the initialization of the student. We follow the general approach [9] where the student is initialized from the teacher's layers. The authors explore the initialization of the student with the first 6 layers, or the final 6 layers of the teacher model, but concluded that using alternating layers of the teacher offers the best initialization, i.e., layer n of the student is initialized from layer $2n - 1$ of the teacher and we, therefore, adopt an identical initialization.

4. Experiments

For the experiments we build upon the pilot experiments discussed in Singh and Lefever [8] using mBERT [1], as well as experiment with another state-of-the-art multilingual teacher, i.e., XLM-RoBERTa [31]. We name our approach *Eliquare* which is the Latin word for 'distillation, filtering or refining'. For both setups, the student (*Eliquare*) is initialized from the given teacher (mBERT or XML-RoBERTa) using the $2n - 1$ approach described in Section 3.

We experiment with six target languages for distillation: French, Dutch, Hindi, Hebrew, Slovene, and Swahili. As can be derived from Table 2, these languages have been selected because they are varied in terms of typology (they all belong to different language groups), script, and resources available (expressed in the number of available Wikipedia pages; for reference, English has 57.29 million Wikipedia pages). Based on this latter column, we consider Dutch and French as representative of high-resourced languages, Hindi and Hebrew as moderately-resourced languages and Swahili and Slovene as low-resourced in our experiments and analyses.

Table 2. An overview of the target languages used for distillation, their genus, scripts and available Wikipedia pages (in millions).

Language	Genus	Script	Wikipedia Pages (in Millions)
French	Romance	Latin	12.318 M
Dutch	Germanic	Latin	4.495 M
Hebrew	Semitic	Ktav Ashuri	1.380 M
Hindi	Indic	Devanagari	1.215 M
Slovene	Slavic	Latin	0.444 M
Swahili	Bantu	Latin	0.155 M

The same Wikipedia dumps of these target languages are used as distillation data in order to construct the *Eliquare* student models with the basic distillation setup. For each language, we obtain the latest Wikipedia XML dumps and pre-process them for MLM, with a masking probability of 0.15 and word masking, word replacement, and unchanged word proportions of 0.8, 0.1, and 0.1, respectively. We also employ the MLM smoothing parameter (set to 0.7) to emphasize masking of less frequent words. Next, the pre-processed data is split into two parts for training and validation with a 90:10 split. All students are trained for 10 iterations over this processed data, using a starting learning rate of 5×10^{-4}. As learning from larger batches works better for distillation, we opted for a batch size of 32 (8 per device) and performed gradient accumulation for 50 steps (effective batch size of $32 * 50 = 1600$). We use the Adam optimizer with an ϵ of 1×10^{-6}. The position embeddings in XLM-RoBERTa are frozen to save some computing time. We store the student model after every epoch and use the version with the best distillation loss on the held-out validation set for the evaluation step.

For the evaluation step, a logical choice could be to look at perplexity and validation loss. However, these are not the best metrics to assess the overall language understanding of an LLM, since they focus on evaluating the Masked Language Modeling objective, rather than general language understanding. Instead, we decided to assess the six monolingual students by fine-tuning them for different language-specific downstream tasks. For each target language, two downstream tasks have been selected, as summarized in Table 3. One task each time requires higher-level (semantic) sentence understanding (such as Sentiment Analysis or News Classification) while the other is highly syntax-dependent (such as Part-of-Speech Tagging). Please note that for the two under-resourced target languages Slovene and Swahili, it was not always possible to find available datasets for these tasks. In those cases, we fell back to the task of named entity recognition or NER, which can be perceived as a task requiring both semantic (which entities do these refer to in the real world) and syntactic (often named entities consists of more than one token) understanding.

Table 3. An overview of the two downstream tasks that have been used to evaluate the language understanding of the *Eliquare* student models for each target language.

Language	Evaluation Task 1	Evaluation Task 2
French	Sentiment Analysis	UD POS
Dutch	Sentiment Analysis	UD POS
Hebrew	Sentiment Analysis	UD POS
Hindi	News Genre Classification	UD POS
Slovene	NER	UD POS
Swahili	News Genre Classification	NER

For Task 1 we employed Sentiment Analysis data from various sources for three languages: Le et al. [32] for French, Van der Burgh and Verberne [33] for Dutch, and

Amram et al. [34] for Hebrew. For Hindi and Swahili we relied on News Genre Classification data from Hindi2Vec (https://github.com/NirantK/hindi2vec, accessed on 1 January 2023), comprising 14 news classes, and from SNCD (https://huggingface.co/datasets/swahili_news, accessed on 1 January 2023) comprising 6 news classes in Swahili. Due to the unavailability of a suitable semantic sentence-level task for Slovene, we used NER data from Rahimi et al. [35] as an alternative. For Task 2 we relied on the Universal Dependencies (https://universaldependencies.org, accessed on 1 January 2023) (UD) project, which comprises treebanks with unified POS-tagged data for French (GSD), Dutch (Lassy-small), Hebrew (HTB), Hindi (HDTB), and Slovene (SSJ). Since there is no UD (or other) treebank publicly available for Swahili, we fell back to NER and used NER data from the Masakhane initiative [36].

We train the student of the respective language individually for each downstream task for 10 epochs with a starting learning rate of 5×10^{-5} with a decay of 0.01 after 500 warmup steps. We select the best validation model (train-validation-test splits are used as provided by the datasets; however, when this is not provided, an 80-10-10 split is used). All tasks are evaluated using F1-score, except task 1 for Dutch (DBRD) which is evaluated with accuracy to allow comparisons with the upper bound.

The results of these experiments are presented in Table 4. Each time we compare the performance of our student models (*Eliquare*-mBERT and *Eliquare*-XLM) to a similarly sized reference, namely distilmBERT which serves as our baseline. Moreover, a comparison is made with the two teacher models and we also represent the upper bound (row in gray) which is each time based on monolingual transformers of the same size as the standard, *BERT-base-uncased* for English. These upper bounds, therefore, are of much larger sizes and trained with multitudes more monolingual data for the target language, while also having a significantly larger and specialized vocabulary for the script in question. The best results per transformer algorithm (BERT/RoBERTa) for each language and task are indicated in bold. From the table, we can observe that the *Eliquare* models often perform similarly or in some cases even better than the respective teachers, i.e., mBERT and XLM, which are much larger in size. The statistical significance of *Eliquare-mBERT*'s improvement over the teacher mBERT was validated using the Wilcoxon Signed-Rank (Left-Tailed) Test ($p = 0.017$) (Statistically significant if $p < 0.05$). Moreover, in a number of low-resourced settings, specifically, for Hebrew (task 2), Slovene (task 1), and Hindi (tasks 1 and 2) the students sometimes even outperform the upper bound. The monolingual performance of the *Eliquare* student models further emphasizes the added value of language-specific distillation since in low-resource settings (Slovene and Swahili) the much more efficient and sustainable student models are able to compete in performance with their larger upper-bounds trained on extensive amounts of monolingual data, making them a better choice for deployment in practical scenarios. It is important to stress the advantages of *Eliquare* students for sustainability and efficiency. The base *Eliquare* student after *vocabulary reduction* (Section 5) has 66 million parameters, which is 2.5 times less than mBERT, and 2 times less than distilmBERT, while also having a significantly faster inference speed of 0.066 s, compared to mBERT's 0.384 s (single V100 GPU with a batch size of 32).

Table 4. A summary of the results for the basic distillation setups for all six languages with mBERT and XLM-RoBERTa as the teachers, respectively, for *Eliquare*-mBERT and *Eliquare*-XLM.

	Task 1	**Task 2**	**Task 1**	**Task 2**	**Task 1**	**Task 2**
	French		**Hebrew**		**Slovene**	
Upper bound	0.9338	0.9818	0.8871	0.9620	0.9410	0.9902
mBERT	0.8923	**0.9795**	0.8512	0.9681	0.9326	0.9791
distilmBERT	0.8773	0.9790	0.8391	0.9597	0.9268	0.9790
Eliquare-mBERT	**0.8952**	0.9792	**0.8567**	**0.9705**	**0.9365**	**0.9822**
XLM-RoBERTa	**0.9273**	**0.9827**	0.8415	**0.9711**	0.9409	0.9865
Eliquare-XLM	0.8938	0.9809	**0.8494**	0.9665	**0.9421**	**0.9865**

Table 4. *Cont.*

	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2
	Dutch		**Hindi**		**Swahili**	
Upper bound	0.9300	0.9630	0.2553	0.9208	0.9090	0.8850
mBERT	**0.9033**	0.9623	0.4744	0.9666	0.8689	0.8490
distilmBERT	0.8812	0.9607	0.4555	0.9597	0.8666	0.8452
Eliquare-mBERT	0.8970	**0.9625**	**0.5066**	**0.9683**	**0.8701**	**0.8632**
XLM-RoBERTa	**0.9240**	**0.9655**	0.4242	0.9754	**0.8816**	**0.8676**
Eliquare-XLM	0.9060	0.9625	**0.4264**	**0.9765**	0.8777	0.8633

These results clearly demonstrate, that even with a vanilla distillation setup, it is possible to obtain better monolingual models for low-resourced languages from a multilingual teacher. In the next sections, we further explore the changes that could be made to the vanilla setup to make the language-specific application of distillation even more viable.

5. Vocabulary Manipulation for mBERT

While the *Eliquare* distilled student models achieve results on par with their respective multilingual teacher models (see Table 4), there are still issues that need to be addressed when using them in a monolingual setting. The most vital of these issues pertains to the multilingual vocabulary of these huge multilingual LLMs.

As visualized in Figure 1, the vocabulary of multilingual models, in this case mBERT, heavily favors Latin-based languages, while having only a meager few thousand sub-words for large language groups such as Indic (6545 to be exact, which can be derived from the circa 12 included languages in mBERT from the Indian sub-continent) and Cyrillic (10 languages and 13,782 sub-words in mBERT). Having a smaller vocabulary in these languages thus means less diverse sub-words which inevitably results in some semantically meaningless alphabet-based tokens in the vocabulary, such as *##a*, etc.

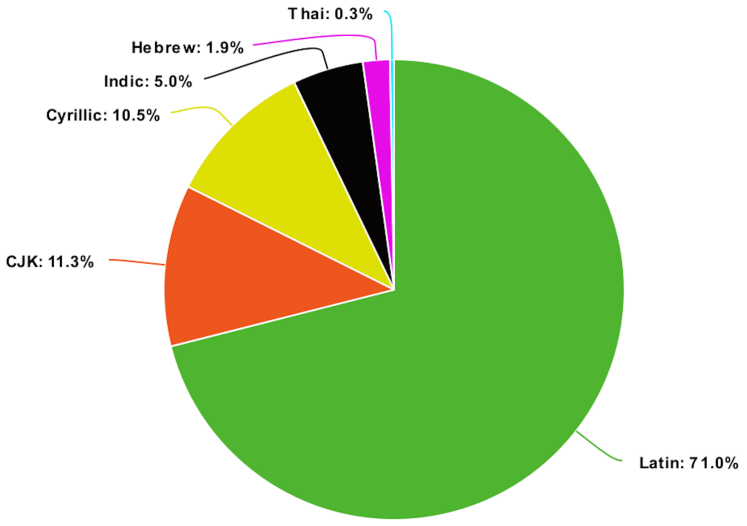

Figure 1. A summary of the distribution of vocabulary for 6 different scripts in mBERT (where *CJK* stands for *Chinese–Japanese–Korean*).

As an example, Figure 2 represents the tokenization of long words in English, a similar high-resourced language (Dutch), a medium-resourced language (Hindi), and a low-resourced language (Farsi). We compare the tokenization by mBERT's WordPiece tokenizer to that of a monolingual model in the respective language. As illustrated in the figure the tokenization is consistent between mBERT and the monolingual model for English, on average having a size of around three characters per sub-word. However, this changes as we go down the resource ladder. For Dutch, some sub-words are only two characters

long, especially sub-words that do not have much semantic meaning attached to them. However, the final two sub-words for Dutch still have 4–5 characters and allow them to have some abstract sense associated with them. Finally, for the final two examples in Hindi and Farsi, mBERT ends up breaking down the word into each individual character, whereas the monolingual model considers the example as an independent whole sub-word.

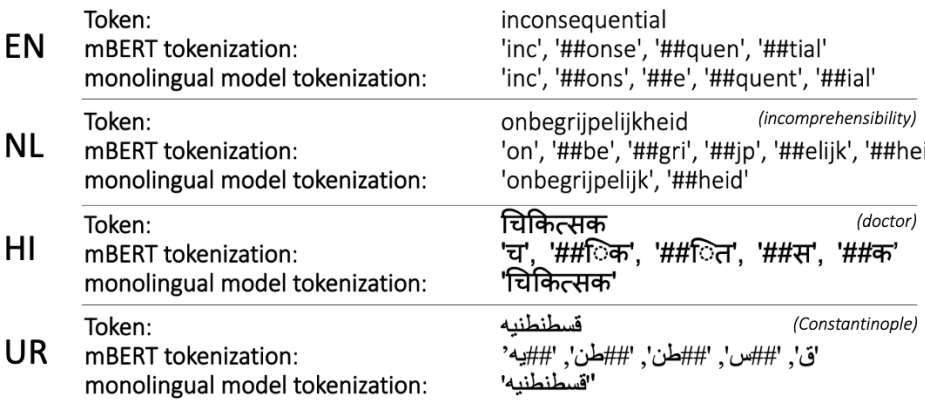

Figure 2. Examples of tokenization for long words in English (EN), Dutch (NL), Hindi (HI), and Urdu (UR), to show the contrast between the obtained sub-words from mBERT and a state-of-the-art monolingual transformer for the respective language.

These tokenization issues, combined with the poor overall representation of low-resourced languages in the vocabulary space are a motivation to investigate strategies to alter multilingual vocabulary for use in a monolingual target language setting, while XLM-RoBERTa suffers from many of the same issues, the Word-Piece Tokenizer of mBERT allows some flexibility to alter the vocabulary even after pre-training, while the Byte-Pair Encoding (BPE) Tokenizer of XLM-RoBERTa is more rigid and does not allow vocabulary deletions/additions as easily. This is why for this and the next section we only experiment with mBERT to alleviate this vocabulary issue. However, we do hope to transfer the methodologies to XLM-RoBERTa in future work.

Hypothetically, two stages can be discerned when building a monolingual student with the ideal low-resourced vocabulary. Firstly, mBERT can be purged of any additional sub-words that may not be needed for a particular target language. We will call this the *VocabReduce* step. Two alternate methodologies can be used for this step. On the one hand, the distillation can work identically to the basic setup, and the vocabulary can be reduced post-distillation directly from the student by removing unnecessary tokens (as proposed by Abdaoui et al. [37]. On the other hand, vocabulary can be reduced pre-distillation, i.e., directly from the teacher. By purging additional sub-tokens from the teacher, we ensure that the student does not initialize the vocabulary for the additional sub-words. In this step pre-distillation reduction has a significant advantage over post-distillation as the distillation can go significantly faster. This is because the sizes of both the student and the teacher are reduced significantly beforehand, thus reducing the number of parameters and by extension the computing time for each iteration.

For the second stage additional richer sub-words could be input to the target language to force the tokenization to not result in meaningless character-based sub-words *VocabAmp*. The vocabulary setups for all the discussed methodologies are summarized in Figure 3. It should be noted that the *VocabAmp* step is more complex as in order to learn additional representations for non-existent sub-words one needs to rely exclusively on external data since the teacher does not possess representations for these missing sub-words. Moreover, given a mismatch between the logits of the teacher and the student, the standard distillation loss cannot be computed since it relies on the divergence between the teacher and student logits. Due to these additional challenges, we consider *VocabAmp* beyond the scope of this work and focus on *VocabReduce*.

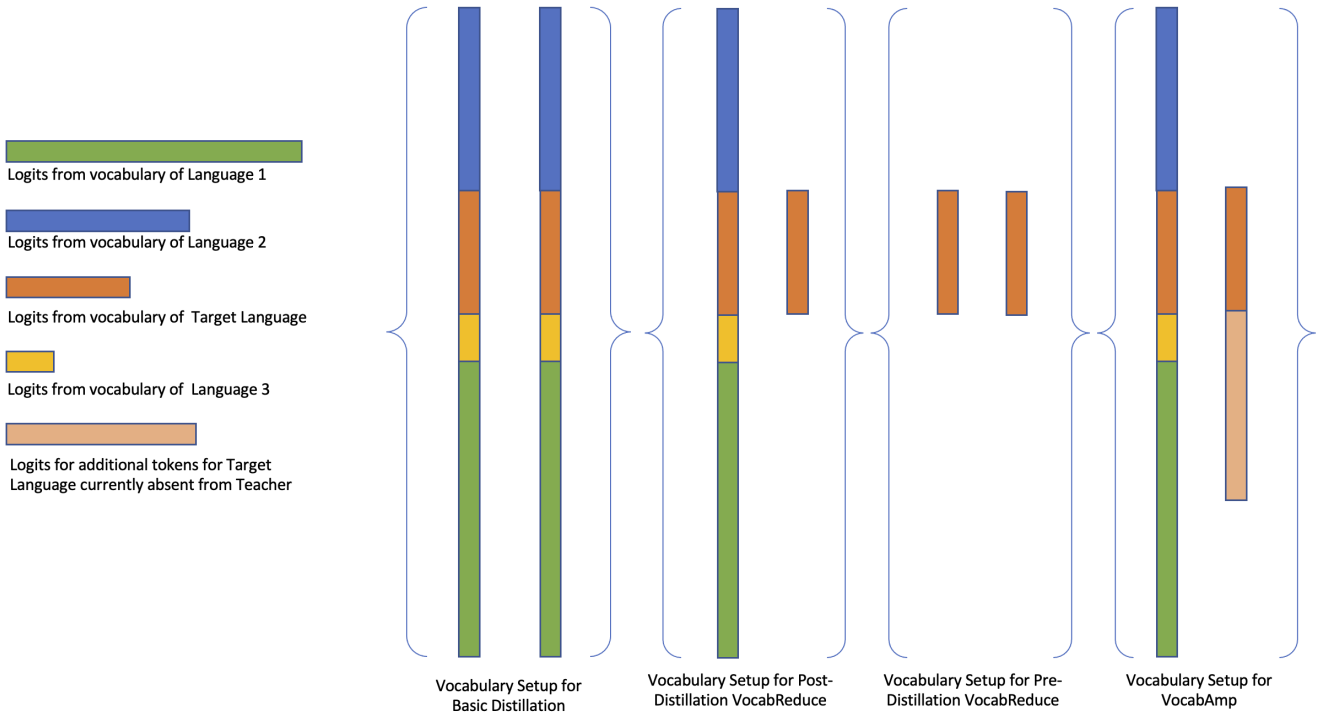

Figure 3. A visual representation of the different vocabulary setups for basic distillation, post-distillation VocabReduce, pre-distillation VocabReduce, and VocabAmp.

We perform experiments with pre- and post-distillation *VocabReduce* for all six languages with mBERT as the teacher. We initialize a list of sub-words for the target language that we would like to retain, by tokenizing the respective Wikipedia dump and selecting sub-words that exist in at least 0.05% of the sentences. We then proceed to reinitialize the transformer's embedding layer and tokenizer so only the selected sub-words are retained. For pre-distillation we apply this technique directly to the teacher, while in post-distillation we apply it to the student after the distillation process. Table 5 shows the result of the experiments for the two tasks for each of the six target languages, while post-distillation results in near-identical performance to the basic distillation setup due to the reduction only taking place afterwards. Pre-distillation comes with minor variance, sometimes better and sometimes worse compared to post-distillation; however, it is consistently faster to train due to the significant reduction in the model's embedding layer sizes. Since the performance difference is barely noticeable, pre-distill VocabReduce should be the go-to methodology due to the additional advantages it comes with.

Table 5. An overview of results for the more advanced post- and pre-distill *VocabReduce* techniques for all six languages.

	Task 1	**Task 2**	**Task 1**	**Task 2**	**Task 1**	**Task 2**
	French		**Hebrew**		**Slovene**	
Post-distill VocabReduce	**0.8952**	0.9788	**0.8565**	0.9705	0.9362	0.9819
Pre-distill VocabReduce	0.8881	**0.9814**	0.8523	**0.9735**	**0.9495**	**0.9849**
	Dutch		**Hindi**		**Swahili**	
Post-distill VocabReduce	0.8964	0.9622	**0.5061**	**0.9682**	0.8691	**0.8629**
Pre-distill VocabReduce	**0.9042**	**0.9634**	0.4979	0.9667	**0.8788**	0.8522

6. Analysis for Low-Resourced Settings

While the basic distillation setups seem to be quite robust in obtaining comparable performance to the multilingual counterparts and in some cases even comparable to the respective upper-bounds, we look into further adaptations that can be made to make distillation setups more suitable for the low-resourced setting. To this end, we perform an ablation study with two vital parts of the distillation pipeline:

1. **Loss Components:** we attempt to find the most and least impactful components of the three-fold loss function to better tune loss weights for low-resourced settings.
2. **Softmax Temperature**: while softening the distribution with a temperature of 2.0 is standard practice in most distillation settings, we dig deeper and see if hardening or further softening can have an impact in the low-resourced setting.

To study the impact of these two variables, we perform additional experiments for the two low-resourced languages: Slovene and Swahili. For the baseline setup, we use the distilled student from the previous section with pre-distillation vocabulary reduction using mBERT as the teacher.

For the first ablation study we thus experiment with the three-fold loss function. The results are presented in Table 6 where the baseline scores (row 1) represent the setup from Section 4 with losses weighted with alpha values of 5.0, 1.0, and 2.0, respectively. The second row gives a general indication of the performance when all losses are weighted equally, while the next three rows show the impact of the individual loss components by removing them from the setup. We notice a drop in performance. Figure 4 also provides a visual intuition of the trends by visualizing this drop in performance in the even weights setting (row 2). The figure demonstrates that each loss component is vital to the setup, which is in line with the consistent drop in performance (rows 3–5) when removing any of the losses. It is also possible to infer from the figure that L_{mlm} is the most pivotal component of the loss function. This is quite an intuitive finding since the student models often perform better than mBERT, their teacher. For these students to learn information missing from the teacher, they would have to rely on knowledge that is not present in the teacher but comes from external sources. In that respect L_{mlm} is the only component able to provide such an external signal. This especially holds in a low-resourced setting, where mBERT's signals may not always be reliable.

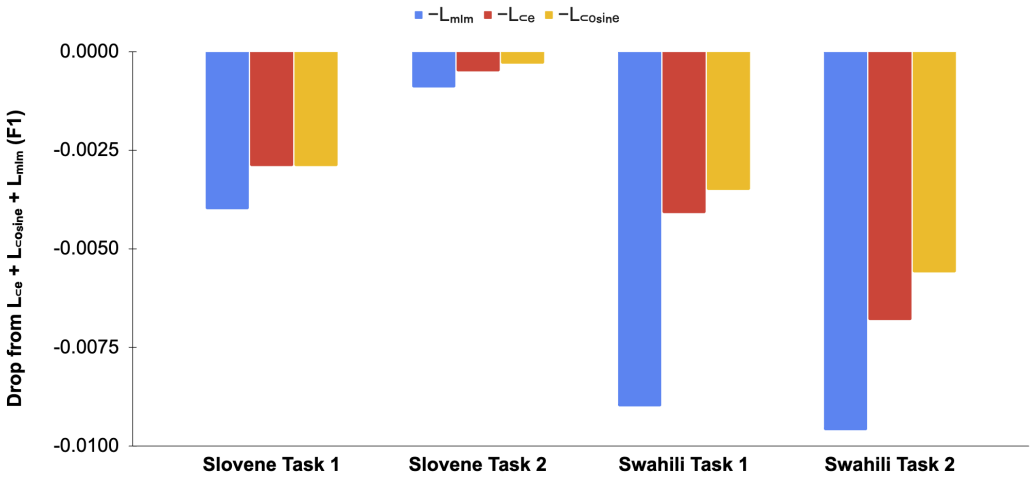

Figure 4. A representation of the results from Table 6 to visualize the drop in performance (in F1-score) from the equal weights setup of $L_{CE} + L_{cosine} + L_{mlm}$ compared to the baseline.

Table 6. Results for Slovene and Swahili for both tasks for the first ablation study. The first row refers to the results of the baseline from Section 4. The second row represents losses with equal weights to set up a comparison for each of the next three experiments where one of the losses is removed from the setup one by one.

Setting	Slovene		Swahili	
	Task 1	**Task 2**	**Task 1**	**Task 2**
$5L_{CE} + 1L_{cosine} + 2L_{mlm}$	**0.9495**	**0.9849**	**0.8788**	0.8522
$L_{CE} + L_{cosine} + L_{mlm}$	0.9440	0.9846	0.8766	**0.8558**
$L_{CE} + L_{cosine}$	0.9400	0.9837	0.8676	0.8492
$L_{cosine} + L_{mlm}$	0.9411	0.9841	0.8725	0.8520
$L_{CE} + L_{mlm}$	0.9411	0.9843	0.8731	0.8532

For our second ablation study, we experiment with the softmax temperature (τ). The results are presented in Table 7, while a τ of 2.0 was used in the baseline experiments in Section 4, four additional experiments have been performed. For two the distribution was further softened with a τ of 3.0 and 4.0, one uses the unchanged logits from the teacher ($\tau = 1.0$) and for another the distribution was hardened ($\tau = 0.5$), while at first sight, the other setups seem to be only marginally deficient, the baseline setup with a τ of 2.0 is consistently better. This indicates that further softening or hardening the logits does not benefit the student specifically in a low-resourced setting.

Table 7. Results for Slovene and Swahili for both tasks illustrating the impact of either softening or hardening the softmax temperature (τ). $\tau = 2.0$ refers to the baseline setup from Section 4.

Temperature (τ)	Slovene		Swahili	
	Task 1	**Task 2**	**Task 1**	**Task 2**
0.5	0.9398	0.9843	0.8740	0.8451
1.0	0.9421	0.9844	0.8736	0.8468
2.0	**0.9495**	**0.9849**	**0.8788**	**0.8522**
3.0	0.9431	0.9842	0.8753	0.8430
4.0	0.9427	0.9840	0.8737	0.8492

Figure 5 elaborates on this finding, as it shows the drop in performance from the peak F1 score at 2.0. While there are some anomalies, it seems to be the case that the further we move from the optimal τ of 2.0, the worse the performance becomes. It is also important to note that tasks such as POS-tagging for Slovene, seem to be quite robust to drops in performance with changes in τ. However, this might be only because the dataset is comparatively easier and performance might already be quite saturated with extremely high scores of the order of 0.984.

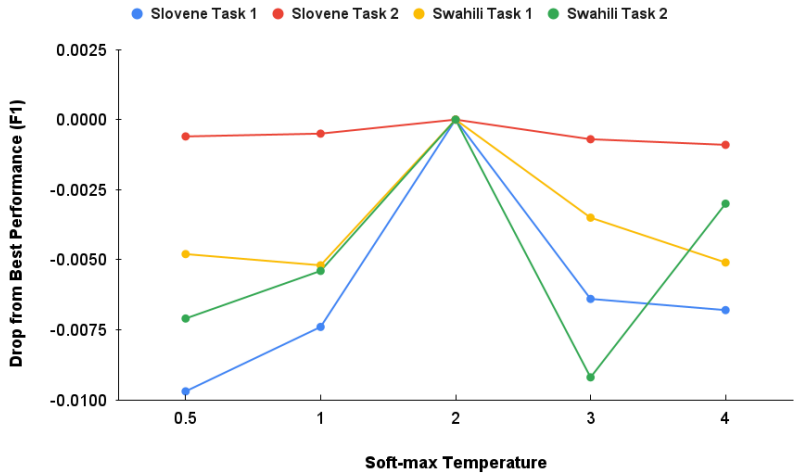

Figure 5. A visualization of the results in Table 7 expressing the relative drops in F1-score from the best value of softmax temperature, $\tau = 2.0$.

7. Conclusions

In this work, we have further explored and improved upon the novel language distillation methodology first introduced in Singh and Lefever [8], where it was tested for mBERT [1]. In this research, we have extended the approach to the more robust and state-of-the-art XLM-RoBERTa [31] and demonstrated its efficacy. Similarly to the language-distillation systems developed from mBERT, the *Eliquare* students of XLM-RoBERTa are able to produce consistent student models for six languages. These languages were carefully selected to account for as much variation as possible with regard to their typologies, language families and available resources. We considered Dutch and French as representative of high-resourced languages, Hindi and Hebrew as moderately resourced languages, and Swahili and Slovene as low-resourced languages. The experimental results confirmed that language-distillation is viable, especially in low-resourced settings, and the resulting students were often able to outperform the teacher multilingual models while being up to four times smaller and six times faster for inference than their respective teachers.

The objective of this research was to further progress research in low-resourced languages, in particular by creating systems for these languages building on existing large multilingual models. This area of research was explored further by looking into the manipulation of the vocabulary of the resulting student models. Two different strategies were proposed to reduce a multilingual vocabulary into a monolingual one as part of the distillation process. We showed that *pre-distillation VocabReduce* is a consistently better strategy since it performs just as well and saves computing time over the alternative, *post-distillation VocabReduce*.

In addition, we also explored the impact of the different loss components on the student of the two low-resourced languages. We discovered that L_{mlm} is the most impactful component of the triplet loss. However, all losses contribute to the performance and the ablation of any component results in a drop in performance.

Finally, we also investigated optimal softmax temperatures in the low-resourced setting and concluded that the default values of $\tau = 2.0$ are optimal, further softening or hardening of the logits results in a drop in performance.

In future work, we would like to venture into more advanced distillation setups described in Section 2, such as TinyBERT [21] and MobileBERT [38], with additional loss components such as Feature Map Transfer and Attention Transfer. We also aim to explore alternate teacher–student setups with multiple teachers, and the construction of bilingual students for two typologically related languages. A logical next step then will be to research strategies for *VocabAmp*, while also modifying the *VocabReduce* technique for application to XLM-RoBERTa.

Author Contributions: Conceptualization, P.S., O.D.C. and E.L.; methodology, P.S., O.D.C. and E.L.; software, P.S.; validation, P.S., O.D.C. and E.L.; formal analysis, P.S.; investigation, P.S.; resources, P.S.; data curation, P.S.; writing—original draft preparation, P.S.; writing—review and editing, P.S., O.D.C. and E.L.; visualization, P.S.; supervision, O.D.C. and E.L.; project administration, E.L.; funding acquisition, E.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; Volume 1, pp. 4171–4186.
2. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747.
3. Martin, L.; Muller, B.; Ortiz Suárez, P.J.; Dupont, Y.; Romary, L.; de la Clergerie, É.; Seddah, D.; Sagot, B. CamemBERT: A Tasty French Language Model. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7203–7219.
4. Delobelle, P.; Winters, T.; Berendt, B. RobBERT: A Dutch RoBERTa-based Language Model. *arXiv* **2020**, arXiv:cs.CL/2001.06286.
5. Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 July 2021; pp. 483–498. https://doi.org/10.18653/v1/2021.naacl-main.41.
6. Chi, Z.; Huang, S.; Dong, L.; Ma, S.; Zheng, B.; Singhal, S.; Bajaj, P.; Song, X.; Mao, X.L.; Huang, H.; et al. XLM-E: Cross-lingual Language Model Pre-training via ELECTRA. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Volume 1, pp. 6170–6182. https://doi.org/10.18653/v1/2022.acl-long.427.
7. Hu, J.; Ruder, S.; Siddhant, A.; Neubig, G.; Firat, O.; Johnson, M. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation. In Proceedings of the 37th International Conference on Machine Learning, Virtual Event, 13–18 July 2020; Volume 119, pp. 4411–4421.
8. Singh, P.; Lefever, E. When the Student Becomes the Master: Learning Better and Smaller Monolingual Models from mBERT. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 29–31 July 2022; pp. 4434–4441.
9. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
10. Sachdeva, N.; McAuley, J. Data Distillation: A Survey. *arXiv* **2023**. https://doi.org/10.48550/ARXIV.2301.04272.
11. Hinton, G.E.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
12. Buciluǎ, C.; Caruana, R.; Niculescu-Mizil, A. Model Compression. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 535–541. https://doi.org/10.1145/1150402.1150464.
13. Tang, Z.; Wang, D.; Zhang, Z. Recurrent Neural Network Training with Dark Knowledge Transfer. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 5900–5904. https://doi.org/10.1109/ICASSP.2016.7472809.
14. Xu, J.; Wang, P.; Yang, H.; L'opez, A.M. Training a Binary Weight Object Detector by Knowledge Transfer for Autonomous Driving. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2379–2384.
15. Mun, J.; Lee, K.; Shin, J.; Han, B. Learning to Specialize with Knowledge Distillation for Visual Question Answering. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 8092–8102.
16. Zhou, C.; Neubig, G.; Gu, J. Understanding Knowledge Distillation in Non-autoregressive Machine Translation. *arXiv* **2020**, arXiv:1911.02727.

17. Fetahu, B.; Veeragouni, A.; Rokhlenko, O.; Malmasi, S. Distilling multilingual transformers into CNNs for scalable intent classification. In Proceedings of the EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022.
18. Chen, Y.C.; Gan, Z.; Cheng, Y.; Liu, J.; Liu, J. Distilling Knowledge Learned in BERT for Text Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7893–7905. https://doi.org/10.18653/v1/2020.acl-main.705.
19. Tang, R.; Lu, Y.; Liu, L.; Mou, L.; Vechtomova, O.; Lin, J. Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. *arXiv* **2019**, arXiv:1903.12136.
20. Sun, S.; Cheng, Y.; Gan, Z.; Liu, J. Patient Knowledge Distillation for BERT Model Compression. 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), Hong Kong, China, 3–7 November 2019; pp. 4323–4332. https://doi.org/10.18653/v1/D19-1441.
21. Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; Liu, Q. TinyBERT: Distilling BERT for Natural Language Understanding. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372.
22. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Brussels, Belgium, 1 November 2018 2018; pp. 353–355. https://doi.org/10.18653/v1/W18-5446.
23. Üstün, A.; Berard, A.; Besacier, L.; Gallé, M. Multilingual Unsupervised Neural Machine Translation with Denoising Adapters. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 6650–6662. https://doi.org/10.18653/v1/2021.emnlp-main.533.
24. Libovický, J.; Rosa, R.; Fraser, A. On the Language Neutrality of Pre-trained Multilingual Representations. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 1663–1674. https://doi.org/10.18653/v1/2020.findings-emnlp.150.
25. Conneau, A.; Wu, S.; Li, H.; Zettlemoyer, L.; Stoyanov, V. Emerging Cross-lingual Structure in Pretrained Language Models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6022–6034. https://doi.org/10.18653/v1/2020.acl-main.536.
26. Pires, T.; Schlinger, E.; Garrette, D. How Multilingual is Multilingual BERT? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4996–5001. https://doi.org/10.18653/v1/P19-1493.
27. Wu, S.; Dredze, M. Are All Languages Created Equal in Multilingual BERT? *arXiv* **2020**, arXiv:2005.09093.
28. Peng, B.; Jin, X.; Liu, J.; Zhou, S.; Wu, Y.; Liu, Y.; Li, D.; Zhang, Z. Correlation Congruence for Knowledge Distillation. *arXiv* **2019**, arXiv:1904.01802.
29. Lee, S.; Song, B.C. Graph-based Knowledge Distillation by Multi-head Attention Network. *arXiv* **2019**, arXiv:1907.02226.
30. Turc, I.; Chang, M.; Lee, K.; Toutanova, K. Well-Read Students Learn Better: The Impact of Student Initialization on Knowledge Distillation. *arXiv* **2019**, arXiv:1908.08962.
31. Lample, G.; Conneau, A. Cross-lingual Language Model Pretraining. *arXiv* **2019**, arXiv:1901.07291.
32. Le, H.; Vial, L.; Frej, J.; Segonne, V.; Coavoux, M.; Lecouteux, B.; Allauzen, A.; Crabbé, B.; Besacier, L.; Schwab, D. FlauBERT: Unsupervised Language Model Pre-training for French. *arXiv* **2019**. https://doi.org/10.48550/ARXIV.1912.05372.
33. van der Burgh, B.; Verberne, S. The merits of Universal Language Model Fine-tuning for Small Datasets—A case with Dutch book reviews. *arXiv* **2019**, arXiv:1910.00896.
34. Amram, A.; Ben David, A.; Tsarfaty, R. Representations and Architectures in Neural Sentiment Analysis for Morphologically Rich Languages: A Case Study from Modern Hebrew. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 21–25 August 2018; pp. 2242–2252.
35. Rahimi, A.; Li, Y.; Cohn, T. Massively Multilingual Transfer for NER. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 151–164.
36. Adelani, D.I.; Abbott, J.; Neubig, G.; D'souza, D.; Kreutzer, J.; Lignos, C.; Palen-Michel, C.; Buzaaba, H.; Rijhwani, S.; Ruder, S.; et al. MasakhaNER: Named Entity Recognition for African Languages. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 1116–1131. https://doi.org/10.1162/tacl_a_00416.
37. Abdaoui, A.; Pradel, C.; Sigel, G. Load What You Need: Smaller Versions of Multilingual BERT. In Proceedings of the SustaiNLP/EMNLP, Online, 20 November 2020.
38. Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; Zhou, D. MobileBERT: A Compact Task-Agnostic BERT for Resource-Limited Devices. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–10 July 2020; pp. 2158–2170. https://doi.org/10.18653/v1/2020.acl-main.195.