# RobBERTje: a Distilled Dutch BERT Model

**Pieter Delobelle**[*] PIETER.DELOBELLE@CS.KULEUVEN.BE
**Thomas Winters**[*] THOMAS.WINTERS@CS.KULEUVEN.BE
**Bettina Berendt**[**] BETTINA.BERENDT@CS.KULEUVEN.BE

[*] *Department of Computer Science; Leuven.AI, KU Leuven, Belgium*

[**] *TU Berlin and Weizenbaum Institute, Germany; KU Leuven, Belgium*

## Abstract

Pre-trained large-scale language models like BERT have gained a lot of attention thanks to their outstanding performance on a wide range of natural language tasks. However, due to their large number of parameters, they are resource-intensive both to deploy and to finetune. As such, researchers have created several methods for distilling language models into smaller ones to increase efficiency, with a small performance trade-off. In this paper, we create a distilled version of the state-of-the-art Dutch RobBERT model, and call it RobBERTje. We found that while shuffled datasets perform better, concatenating sequences from the non-shuffled dataset to improve the dataset sequence length distribution before shuffling further improves the performance for tasks with longer sequences, and makes the model faster to distill. Upon comparing architectures, we found that the larger DistilBERT architecture worked significantly better than the BORT hyperparametrization. Since smaller architectures decrease the time to finetune, these models allow for more efficient training and more lightweight deployment of many Dutch downstream language tasks.

## 1. Introduction

Large-scale pre-trained language models like BERT (Devlin et al. 2019) have revolutionized many natural language processing tasks thanks to their outstanding performance on downstream tasks. While the model itself is trained using a Masked Language Modelling (MLM) task in the pre-training phase for gaining linguistic insights, it only requires finetuning on relatively small datasets to get state-of-the-art performance on the required task, such as sentiment analysis, natural language inference and token tagging tasks. However, such language models are usually large and thus slow to inference and difficult to deploy in environments, due to requiring large storage and large amounts of energy (Bender et al. 2021) As such, following the trend of distilling the knowledge from neural network models (Hinton et al. 2015), many types of distillation have been used to extract optimal parameters or extract the knowledge of larger language models into smaller ones (Sanh et al. 2019, de Wynter and Perry 2020, Jiao et al. 2020). These smaller models require fewer resources and time to run, and thus less electricity, at the cost of being slightly less accurate. Such a distillation allows for a favorable trade-off between performance and ease of use.

In this paper, we distill the Dutch BERT model RobBERT v2 (Delobelle et al. 2020), and name it RobBERTje[1]. We perform several distillations using a small unlabeled Dutch dataset and finetune them on several language tasks to find the best processing of the dataset and target architecture hyperparametrizations. The main contributions of this paper are thus (1) evaluating data processing for distillation, which may also extend to better practices for other tasks such as pre-training (2) replicating distillation architectures (3) creating a more light-weight version of RobBERT to enable more efficient finetuning and energy-efficient inferencing of downstream Dutch NLP tasks.

1. Dutch for "Little RobBERT"

## 2. Background

Methods for distilling learned models were initially created for learning a simpler model (student) from a complex ensemble model (teacher) by labeling a large unlabeled dataset using the teacher (Buciluǎ et al. 2006). This was later extended to neural networks, where the student additionally learns the probabilities assigned to the incorrect labels, thus learning to generalize in the same way as the teacher model (Hinton et al. 2015). There are generally two types of distillation that are performed on BERT models, namely task-specific distillation and general distilling, the latter of which still keeps the same properties as the pre-trained model. These are sometimes mixed in a two-stage model, e.g. TinyBERT (Jiao et al. 2020). In task-specific BERT distillation, a pre-trained BERT model is finetuned for a particular task, after which a simpler neural network (e.g. a BiLSTM) learns from the probability estimates of the finetuned model (Tang et al. 2019). General distillation removes some layers from the BERT architecture and aims to train a student model that can then still be finetuned for other downstream tasks, just like its teacher. This is usually achieved by learning the probability distribution for tokens in the masked language task. This was done successfully in DistilBERT, where the model had 40% fewer parameters, and still retained 97% its language understanding while being 60% faster (Sanh et al. 2019). Researchers also found optimal sizes for the student architecture, both experimentally (Turc et al. 2019) and using formal optimal parameter extraction methods, such as BORT (de Wynter and Perry 2020). Most distillation methodologies thus boil down to finding fitting, smaller architectures and procedures for teaching the student model to learn the same distribution over the predictions as the larger, teacher model.

## 3. RobBERTje Distillation Choices

We evaluate several choices when performing distillation to create a distilled version of the Dutch RobBERT model, as this model achieves state-of-the-art results on many downstream Dutch language tasks (Delobelle et al. 2020). More specifically, we experiment with the influence of the training dataset and replicate studies of the DistilBERT and BORT architectures. The RobBERT model is based on the RoBERTa architecture (Liu et al. 2019), an optimized version of the original BERT model, and trained on the Dutch portion of the OSCAR corpus (Ortiz Suárez et al. 2019). As there are many choices to make when distilling a model, we test if it matters if the training corpus is shuffled (§ 3.1), the influence of the length of the training sentences (§ 3.2) and what distillation architecture works best for our model (§ 3.3). We perform these experiments using the first 1GB of the non-shuffled Dutch OSCAR dataset using one Nvidia 1080 Ti. For the MLM perplexity evaluation, we use 50k sequences from the last shard of the non-shuffled dataset.

### 3.1 To Shuffle Or Not To Shuffle?

The OSCAR corpus (Ortiz Suárez et al. 2019) is one of the most used datasets to train large language models. It is constructed by classifying the language of a web-crawled dataset and only provides a shuffled version publicly. While some hypothesized that using a non-shuffled version could allow the model to learn dependencies spanning multiple sequences (Wouts 2020), the order itself is likely not important for pre-training due to using each input sequence individually. Since RoBERTa dropped next-sentence prediction due to being ineffective, models using this optimized training regime also lack dependencies across separate training sequences (Liu et al. 2019). In theory, not shuffling the dataset could hurt the training performance due to less diverse training sequences in every batch. This leads to being less representative of the true gradient over the whole dataset, thus pushing the gradient into less desirable directions. We trained two distilled models (*Shuffled* and *Non-shuffled*) using the DistilBERT regime. We found that while the non-shuffled version has a better MLM head (PLLL) and has similar performance on most fine-tuned tasks, the shuffled version performs significantly better on the DBRD sentiment analysis task (Table 1).

### 3.2 Sequence Merging for Increased Sequence Length

One advantage of the non-shuffled corpus that we aim to identify and evaluate is that it allows for creating longer, meaningful sequences to improve the sequence length distribution. Since most OSCAR sequences are relatively short ($< 40$ tokens, Figure 1), we concatenate sequential lines from the same document into one training sequence. Longer sequences allow later positions to see more data, theoretically improving its performance for downstream tasks with long sequences. In addition, the same data is encoded in fewer sequences, decreasing training time and energy for pre-training and distillation. One downside is that early input positions see relatively less training data.

We created a new dataset from the non-shuffled Dutch OSCAR dataset by randomly merging subsequent sentences with a probability of 50% if they are from the same document. This resulted in a smaller corpus with generally longer sequences (Figure 1). After the sequence merging, we shuffled this corpus, as our first experiment indicated that using a shuffled corpus improves performance.

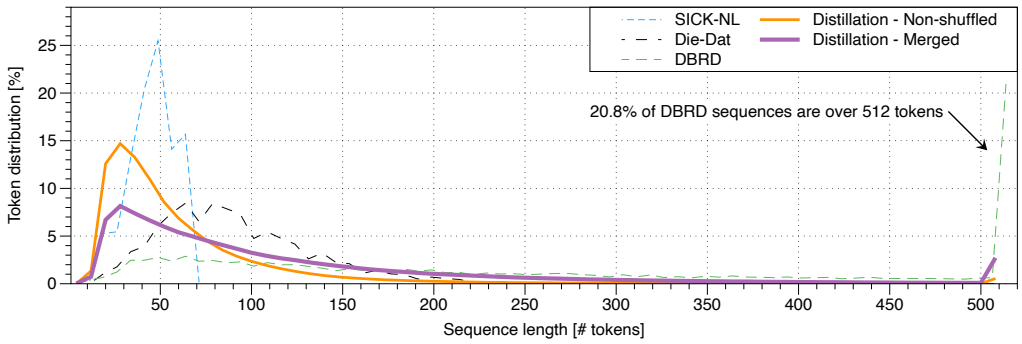


Figure 1: Sequence length distribution of tasks and OSCAR with and without merging sentences.

### 3.3 Target Architecture

There are several choices when it comes to choosing the student architecture for distillation. For all earlier experiments, we used the DistilBERT architecture (Sanh et al. 2019), which roughly halves the number of parameters and keeps the general properties of the teacher (RobBERT). Recently the BORT approach emerged, claiming to have found an optimal parametrization with only 16% the net size of the RoBERTa model (de Wynter and Perry 2020). We aim to replicate this study using the same student architecture hyperparameters, and validate if these parametrizations still hold for this Dutch RoBERTa model. We used our merged sequences dataset and kept the hidden size to 768 to allow hidden distillation, causing the model to be larger than the original BORT.

## 4. Experiments

### 4.1 Benchmark Task Datasets

We evaluated our distilled models on the following language tasks, most of which are described in more detail in the RobBERT paper (Delobelle et al. 2020).

- **DBRD** Sentiment analysis on Dutch book reviews (van der Burgh and Verberne 2019).
- **Die-Dat** Co-reference resolution of *die*/*dat* pronouns on EuroParl (Allein et al. 2020).
- **NER** Named entity recognition on the CoNLL-2002 dataset (Tjong Kim Sang 2002).
- **POS** Part-of-speech tagging using universal dependencies of Lassy (Van Noord et al. 2013).

Table 1: Overview of all pretrained models and benchmark results and number of decoder layers $D$, number of attention heads $A$, hidden size $H$ and intermediate layer size $I$. We report accuracy and 95% CI for all benchmark scores, except NER, which uses the $F_1$ score. Results indicated with $^\dagger$ are sourced from Delobelle et al. (2020). PPPL results for RobBERT are indicative, since we could not guarantee the pre-training data was not seen before.

| | | Hyperparameters | | | | | Benchmark scores | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Data** | **D** | **A** | **H** | **I** | **Params** | **DBRD** | **DIE-DAT** | **NER** | **POS** | **SICK-NL** | **PPPL** |
| Teacher (RobBERT v2) | 39 GB | 12 | 12 | 768 | 3072 | 116 M | $94.4 \pm 1.0^\dagger$ | $99.2 \pm 0.03^\dagger$ | $89.1^\dagger$ | $96.4 \pm 0.4^\dagger$ | $84.2 \pm 1.0$ | *7.76* |
| Non-shuffled (§ 3.1) | 1 GB | 6 | 12 | 768 | 3072 | 74 M | $90.2 \pm 1.2$ | $\mathbf{98.4} \pm 0.1$ | **82.9** | $95.5 \pm 0.4$ | $\mathbf{83.4} \pm 1.0$ | **12.95** |
| Shuffled (§ 3.1) | 1 GB | 6 | 12 | 768 | 3072 | 74 M | $92.5 \pm 1.1$ | $98.2 \pm 0.1$ | 82.7 | $\mathbf{95.6} \pm 0.4$ | $\mathbf{83.4} \pm 1.0$ | 18.74 |
| Merged (§ 3.2) | 1 GB | 6 | 12 | 768 | 3072 | 74 M | $\mathbf{92.9} \pm 1.1$ | $96.5 \pm 0.1$ | 81.8 | $95.2 \pm 0.4$ | $82.8 \pm 1.1$ | 17.10 |
| BORT (§ 3.3) | 1 GB | 4 | 8 | 768 | 768 | 46 M | $89.6 \pm 1.3$ | $92.2 \pm 0.1$ | 79.7 | $94.3 \pm 0.4$ | $81.0 \pm 1.1$ | 26.44 |

- **SICK-NL** Natural language inference dataset (Wijnholds and Moortgat 2021). We added missing punctuation, which significantly improves the performance for BERT models.
- **PPPL** Measures the MLM perplexity on Dutch OSCAR subset, using Salazar et al. (2019).

### 4.2 Evaluation

We evaluated the distilled models on the aforementioned tasks (Table 1). *Shuffled* performs similarly as *Non-shuffled*, except better on sentiment analysis and worse on perplexity. The MLM pseudo-perplexity (PPPL) indicates that *Non-shuffled*'s MLM head best captures the distribution of the test data. The non-merged test data explains its performance gain over the *Merged* model of 32%. However, the PPPL increase of 45% (*Shuffled*) and 104% (*BORT*) are higher than hypothesized and might require further analysis. While the *BORT* version is much smaller and faster (e.g. for SICK-NL, it finetunes 4x faster than RobBERT and 2.2x faster than our merged sequence distillation), it is significantly outperformed by its DistilBERT counterpart (=*Merged*) on all tasks. For the subsequent merging, the results are less clear. As expected, merging data into longer subsequences is advantageous to tasks using long sequences as input (such as DBRD), but less advantageous to tasks that have shorter sequences (such as SICK-NL) (Figure 1). We released these models as RobBERTje on `https://github.com/ipieter/robbertje`.

## 5. Conclusion

In this paper, we created a distilled version of the state-of-the-art Dutch RobBERT model, called RobBERTje. In doing so, we found that using a shuffled dataset is slightly better for distillation. We also found that randomly merging subsequent sequences of the non-shuffled dataset improves the performance of the distilled language model for tasks using longer input sentences. We replicated the BORT approach and found that while the model is much smaller than its DistilBERT counterpart, its performance was significantly less on all tested tasks. The results imply that these new distilled RobBERTje models can effectively be used for making a large number of downstream Dutch natural language processing tasks much more efficient while still achieving close to state-of-the-art results.

#### Acknowledgements

Pieter Delobelle was supported by the Research Foundation - Flanders (FWO) under EOS No. 30992574 (VeriLearn). Pieter Delobelle also received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme. Thomas Winters is supported by the Research Foundation-Flanders (FWO-Vlaanderen, 11C7720N).

## References


Allein, Liesbeth, Artuur Leeuwenberg, and Marie-Francine Moens (2020), Automatically correcting Dutch pronouns" die" and" dat", *Computational Linguistics in the Netherlands Journal* **10**, pp. 19–36.

Bender, Emily M, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell (2021), On the dangers of stochastic parrots: Can language models be too big?, *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623.

Buciluǎ, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil (2006), Model compression, *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541.

de Wynter, Adrian and Daniel J Perry (2020), Optimal subarchitecture extraction for BERT, *arXiv:2010.10499*, ArXiv. https://arxiv.org/abs/2010.10499.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (2020), RobBERT: a Dutch RoBERTa-based Language Model, *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, pp. 3255–3265. https://www.aclweb.org/anthology/2020.findings-emnlp.292.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019), BERT: Pre-training of deep bidirectional transformers for language understanding, *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)*, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015), Distilling the knowledge in a neural network, *arXiv:1503.02531*. https://arxiv.org/abs/1503.02531.

Jiao, Xiaoqi, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu (2020), TinyBERT: Distilling BERT for natural language understanding, *Findings of ACL: EMNLP 2020*.

Liu, Yinhan, 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*. https://arxiv.org/abs/1907.11692.

Ortiz Suárez, Pedro Javier, Benoît Sagot, and Laurent Romary (2019), Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures, *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. https://hal.inria.fr/hal-02148693.

Salazar, Julian, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff (2019), Masked language model scoring, *arXiv preprint arXiv:1910.14659*.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019), DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter, *NeurIPS EMC$^2$ Workshop*.

Tang, Raphael, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin (2019), Distilling task-specific knowledge from bert into simple neural networks, *arXiv preprint arXiv:1903.12136*, ArXiv. https://arxiv.org/abs/1903.12136.

Tjong Kim Sang, Erik F. (2002), Introduction to the conll-2002 shared task: Language-independent named entity recognition, *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, Association for Computational Linguistics, USA, p. 1–4. https://doi.org/10.3115/1118853.1118877.

Turc, Iulia, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019), Well-read students learn better: On the importance of pre-training compact models, *arXiv preprint arXiv:1908.08962v2*. https://arxiv.org/abs/1908.08962.

van der Burgh, Benjamin and Suzan Verberne (2019), The merits of Universal Language Model Fine-tuning for Small Datasets – a case with Dutch book reviews, *arXiv:1910.00896 [cs]*. http://arxiv.org/abs/1910.00896.

Van Noord, Gertjan, Gosse Bouma, Frank Van Eynde, Daniel De Kok, Jelmer Van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste (2013), Large scale syntactic annotation of written dutch: Lassy, *Essential speech and language technology for Dutch*, Springer, Berlin, Heidelberg, pp. 147–164.

Wijnholds, Gijs and Michael Moortgat (2021), SICKNL: A dataset for Dutch natural language inference, *arXiv preprint arXiv:2101.05716*. https://arxiv.org/abs/2101.05716.

Wouts, Joppe Valentijn (2020), Text-based classification of interviews for mental health – juxtaposing the state of the art. https://arxiv.org/abs/2008.01543.