# Comparing Pre-Training Schemes for Luxembourgish BERT Models

**Cedric Lothritz** **Saad Ezzini**
**Christoph Purschke** **Tegawendé F. Bissyandé** **Jacques Klein**
University of Luxembourg
6, rue Coudenhove-Kalergi
L-1359 Luxembourg
{cedric.lothritz, saad.ezzini,christoph.purschke,tegawende.bissyande,jacques.klein} @uni.lu

**Isabella Olariu**
Zortify SA
9, rue du Laboratoire
L-1911 Gare Luxembourg
isabella@zortify.com

**Andrey Boytsov** **Clément Lefebvre** **Anne Goujon**
BGL BNP Paribas
10, rue Edward Steichen
L-2540 Luxembourg
{andrey.boytsov,clement.c.lefebvre,anne.goujon} @bgl.lu

## Abstract

Despite the widespread use of pre-trained models in NLP, well-performing pre-trained models for low-resource languages are scarce. To address this issue, we propose two novel BERT models for the Luxembourgish language that improve on the state of the art. We also present an empirical study on both the performance and robustness of the investigated BERT models. We compare the models on a set of downstream NLP tasks and evaluate their robustness against different types of data perturbations. Additionally, we provide novel datasets to evaluate the performance of Luxembourgish language models. Our findings reveal that pre-training a pre-loaded model has a positive effect on both the performance and robustness of fine-tuned models and that using the German GottBERT model yields a higher performance while the multilingual mBERT results in a more robust model. This study provides valuable insights for researchers and practitioners working with low-resource languages and highlights the importance of considering pre-training strategies when building language models.

**Keywords:** Low-resource languages, Luxembourgish, LuxemBERT, Downstream NLP tasks, Language models, Pre-training, GottBERT, BERT

## 1 Introduction

The introduction of BERT models in 2018 (Devlin et al., 2019) was a crucial milestone for the NLP community. The ability to fine-tune an already pre-trained BERT model mitigated the need for specialised model architectures for given tasks. Despite the emergence of better-performing architectures in recent years, fine-tuning BERT models continues to be a popular approach for numerous NLP tasks in industrial settings.

While highly performing pre-trained BERT models are readily available for widely spoken languages, they are comparably scarce for low-resource languages due to the amount of data necessary to pre-train adequate models. In fact, we determined that the number of languages for which a pre-trained BERT model is available on Huggingface[1] is less than 150, with many of them supported only through multilingual models such as multilingual BERT (mBERT) (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2019). These multilingual models provide a viable alternative, but monolingual models can outperform them if sufficient pre-training data is available, as shown by Wu and Dredze (2020).

Several factors can influence the quality of a language model (LM), such as the size of the pre-training corpus, which can be increased through data augmentation techniques (Hedderich et al., 2020). The configuration of the model architecture can also be varied to improve performance, as highlighted by Wu and Dredze (2020). Another ap-

[1] https://huggingface.co/models



*Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, pages 17–27
September 18–22, 2023. 

proach to enhance the performance of a language model is to choose whether to pre-train the LM from scratch or to pre-load the weights from an existing model and continue the pre-training using data from the target language, as discussed in (Muller et al., 2021). These considerations are important when working with low-resource languages as they can greatly impact the quality of the pre-trained models.

In this study, we focus on Luxembourgish, a low-resource language spoken primarily in Luxembourg by nearly 600 000 people worldwide. We investigate the impact of pre-training a pre-loaded LM versus using pre-training from scratch, as well as the impact of pre-loading a monolingual versus a multilingual pre-trained model.

The contributions of this study are threefold: **(a)** We propose two novel BERT models for the Luxembourgish language that improve on the state of the art. These models are trained on a large corpus of Luxembourgish text and are able to capture the unique characteristics of the language. **(b)** We also present an empirical study on both the performance and robustness of the investigated BERT models. This study compares the models on a set of downstream NLP tasks and evaluates their robustness against different types of data perturbations. **(c)** Additionally, we provide novel datasets to evaluate the performance of Luxembourgish language models. These datasets are specifically designed for the Luxembourgish language and are not available in previous studies, which will be useful for future research in this field.

## 2 Approach

In this section, we describe the creation of the two novel BERT models that we pre-trained for this study: Lb_mBERT and Lb_GottBERT. [2]

### 2.1 Pre-loaded Models

As mentioned in Section 1, we set out to compare pre-loading a multilingual and a monolingual BERT model. Our models of choice are the multilingual mBERT and the German GottBERT model which we pre-train on a corpus of 12 million sentences.

[2]Our final models are available at https://huggingface.co/lothritz/Lb_mBERT and https://huggingface.co/lothritz/Lb_GottBERT

#### 2.1.1 mBERT

Created by Devlin et al. (2019), mBERT is a multilingual BERT model trained on 104 languages. Specifically, the model was pre-trained on Wikipedia articles, including the Luxembourgish Wikipedia, which contained 59 000 articles. mBERT contains 12 transformer blocks, 768 hidden layers, 12 self-attention blocks, and 110 million trainable parameters, as well as a vocab size of 105 879 WordPiece tokens, 100 of which are unused. Our first model uses mBERT as its starting point and is appropriately named Lb_mBERT. We adapt the vocab file by replacing the unused tokens with the 100 most common ones in our pre-training corpus. We then train the model for 10 epochs on the Masked-Language-Modeling task (MLM) with a masking probability of 15%.

#### 2.1.2 GottBERT

Luxembourgish is a West Germanic language originating from a Moselle Franconian dialect (Gilles, 2022). As such, Luxembourgish and German are closely related. Indeed, both languages are similar in terms of vocabulary and structure (Lothritz et al., 2022). Due to these similarities, we choose the German GottBERT model (Scheible et al., 2020) as a pre-loaded model to create Lb_GottBERT. GottBERT was pre-trained on the German part of the OSCAR corpus (Ortiz Suárez et al., 2020) consisting of nearly 459 million sentences. Its vocab file consists of 52 009 WordPiece tokens. As none of these tokens are unused, we cannot modify the vocab file. Similarly to the training of Lb_mBERT, we pre-train the model for 10 epochs on the MLM task with a masking probability of 15% using the same pre-training corpus.

### 2.2 Pre-training Corpus

In order to pre-train our models, we use the corpus built by (Lothritz et al., 2022) which consists of 12 million sentences, 6 million of which are written in Luxembourgish. The used corpus includes data from the Luxembourgish Wikipedia, the Luxembourgish news site rtl.lu, and the Leipzig Wortschatz corpus (Goldhahn et al., 2012). The remaining 6 million consist of augmented data resulting from a novel data augmentation scheme based on partial translation. As Luxembourgish is very closely related to the German language in terms of structure and vocabulary, the authors used a German dataset made up of Wikipedia articles that they partially translate to Luxembourgish. Specifically,

they used a predetermined set of non-ambiguous and common words to translate a significant portion of their supplementary German data to Luxembourgish.

## 3 Experimental Setup

In this section, we list our research questions for this study and describe the setup of experiments we perform to answer these questions. For our experiments, we consider six pre-trained language models finetuned on eight NLP tasks: Part-of-Speech (POS) tagging, Named Entity Recognition (NER), Intent Classification (IC), News Classification (NC), Winograd Natural Language Inference (WNLI), Sentence Negation (SN), Sentiment Analysis (SA), and Recognizing Textual Entailment (RTE). Furthermore, when applicable, we apply four perturbation techniques to our test sets: negation, name replacement, location replacement, and synonym replacement.

### 3.1 Research Questions

We address the following research questions:

**RQ1. Which model yields the highest performance on downstream NLP tasks?** In this research question, we aim to evaluate and compare the performance of different language models on a set of downstream tasks such as news classification, named entity recognition, part-of-speech tagging, etc. The goal is to identify the model that performs the best across all tasks or a specific set of tasks. **RQ2. How robust are the models against data perturbation?** In this research question, we aim to evaluate the robustness of the models against different types of data perturbations, namely: negation, name replacement, location replacement, and synonym replacement. The goal is to understand how well the models can handle these variations in input data and identify the model that is the most robust.

### 3.2 Baseline Models

In this section, we present the various BERT models we investigated for this study. Most of the models were pre-trained on Luxembourgish data. Table 1 shows an overview of the differences between each model.

#### 3.2.1 mBERT & GottBERT

We use the original versions of both mBERT and GottBERT without additional pre-training as two of our baseline models. This allows us to determine the impact of our pre-training corpus on each respective model. While mBERT was partially trained on Luxembourgish Wikipedia articles, GottBERT was trained exclusively on German data. As such, we expect mBERT to yield better performances on the downstream tasks.

#### 3.2.2 LuxemBERT

(Lothritz et al., 2022) published a Luxembourgish BERT model made from scratch trained on the 12 million sentences described in Section2.2. Its architecture is made up of 12 transformer blocks, 768 hidden layers, 12 self-attention blocks, and 110 million trainable parameters, as well as a vocab size of 30 000 WordPiece tokens. It was trained on the MLM task for 10 epochs with a masking probability of 15%. They found that LuxemBERT improved upon mBERT's performance for numerous tasks. Following that, we expect it to outperform both mBERT and GottBERT in most of our experiments.

#### 3.2.3 DA BERT

DA BERT was created by Olariu et al. (2023) and was trained on the same 6 million Luxembourgish sentences as LuxemBERT. Similarly to LuxemBERT, it was pre-trained from scratch, and has a similar architecture to LuxemBERT: 12 transformer blocks, 768 hidden layers, 12 self-attention blocks, and 110 million trainable parameters. The vocab size is also identical with 30 000 tokens. However, contrary to LuxemBERT, the 6 million remaining sentences were not translated from a different language. Instead, they employed classical data augmentation techniques to create more data. Specifically, they replaced words in the original dataset while preserving the original meaning of the original sentences. The word replacements consisted of synonym replacements, named entity replacements, and modal verb replacements. They found that the performance of their new model is similar to that of LuxemBERT. As such, we also expect its performance in our experiments to be comparable to that of LuxemBERT.

### 3.3 Downstream Tasks

For this study, we consider eight downstream tasks. In addition to the five tasks introduced in Lothritz et al. (2022) (POS-tagging, Named Entity Recognition, Intent Classification, News Classification, and WNLI), we also investigate Sentence Negation, the

| | mBERT | GottBERT | LuxemBERT | DA BERT | Lb_mBERT | Lb_GottBERT |
|---|---|---|---|---|---|---|
| Pre-training | NAP | NAP | from scratch | from scratch | from mBERT | from GottBERT |
| Authentic Lb Data | No | No | Yes | Yes | Yes | Yes |
| Translated De Data | No | No | Yes | No | Yes | Yes |
| Augmented Lb Data | No | No | No | Yes | No | No |

Table 1: Differences in pre-training scheme and data for each investigated model. (NAP = no additional pre-training)

Recognizing Textual Entailment task, and Sentiment Analysis, which we describe in the following section.[3]

#### 3.3.1 Part-of-Speech Tagging

Part-of-Speech (POS) tagging task is a classical sequence-to-sequence task. The objective is to categorise each word in a sentence into its correct grammatical class such as noun or verb. This dataset is made up of nearly 5500 sentences from Luxembourgish news articles and words are categorised into 15 different classes (Lothritz et al., 2022).

#### 3.3.2 Named Entity Recognition

Similarly to POS-tagging, Named Entity Recognition (NER) is a common sequence-to-sequence task aimed to detect proper names in text. The raw dataset for this task is the same as the one for POS-tagging, and covers the labels *person*, *geopolitical entity*, *(natural) location*, *organisation*, and *miscellaneous* (Lothritz et al., 2022).

#### 3.3.3 Intent Classification

Intent Classification (IC) is a crucial task for digital assistants and chatbot, concerned with detecting the underlying intent of a user's message. For this study, we use the Banking Client Support dataset introduced in Lothritz et al. (2021). The dataset contains nearly 1000 samples divided into 28 intents for the banking domain.

#### 3.3.4 News Classification

News Classification (NC) is a popular text classification task in NLP. As the name implies, the objective is to categorise news articles into given types of news. This set consists of nearly 10 000 news articles divided into eight labels. (Lothritz et al., 2022)

#### 3.3.5 Winograd Natural Language Inference

Being part of the GLUE benchmark collection (Wang et al., 2018), the Winograd Natural Language Inference (WNLI, Levesque et al., 2012). Given a sentence pair A and B, where A contains at least one pronoun and B replaces the pronoun, the task consists of determining whether or not A entails B. For this study, we use a translated version of the dataset (Lothritz et al., 2022), containing nearly 800 sentence pairs.

#### 3.3.6 Sentence Negation

The Sentence Negation task consists of changing the polarity of a given sentence. Specifically, the objective is to correctly place the word "net"[4] in order to turn the sentence negative. For this task, we only consider sentences that are fewer than 15 words long. The dataset consists of a subset of the Luxembourgish portion of the Leipzig Corpora Collection (Goldhahn et al., 2012)[5], which was not used to pre-train either of our models. We extract all the sentences containing the word "net" and turn them into a labelled dataset accordingly. The resulting training, validation, and test sets contain 33975, 2171, and 10095 sentences, respectively. The word "net" is at position 3 in most sentences (14.52% of the dataset), while it is at position 13 in the fewest cases (0.5%). It is to note, that there are multiple ways to negate sentences in the Luxembourgish language, with slightly different meanings depending on the position of the word "net". As such, a model's prediction may be considered false in our experiments despite producing a correctly negated sentence.

#### 3.3.7 Recognizing Textual Entailment

The Recognizing Textual Entailment (RTE) task was introduced by Haim et al. (2006) and was added to the GLUE benchmark collection (Wang et al., 2018) for evaluating the performance of language models. Given a sentence pair A and B, the objective is to determine whether or not B is entailed by A. As there is currently no Luxembourgish version for this task, we translated the original version to Luxembourgish using the googletrans API.[6] The final dataset contains translation errors,

[3] Our datasets are available at https://github.com/Trustworthy-Software/LuxemBERT

[4] The Luxembourgish word for "not"

[5] https://wortschatz.uni-leipzig.de/en/download/Luxembourgish

[6] https://pypi.org/project/googletrans/

but it is serviceable for our experiments as the data is the same for each of our models. However, we would not advise to use this dataset for commercial use without revising the text. The training, validation, and test sets contain 2490, 277, and 801 sentences, respectively. 51% of the sentence pairs are examples for textual entailment while 49% are not.

#### 3.3.8 Sentiment Analysis

Sentiment Analysis is a classic NLP problem consisting of determining whether a given sentence is positive, negative, or neutral. For this study, we use two different datasets: SA1 and SA2. SA1 is a dataset of Luxembourgish user comments collected from the news website RTL[7] that was manually annotated with the labels *positive*, *negative*, and *neutral*. The training, validation and test sets contain 1293, 188, and 367 samples, respectively. 12% of the samples are labelled positive, 34% negative, and 54% are neutral.[8] SA2 is a subset of the SST-2 dataset (Socher et al., 2013) which we automatically translated to Luxembourgish using Google Translate.

Unlike the SA1 dataset, it has binary labels: *positive* and *negative*. SA2's training, validation, and test sets contain 9646, 872, and 2360 samples, respectively. 55% of the samples are labelled *positive* and 45% negative.

### 3.4 Finetuning Parameters

Devlin et al. (2019) recommends choosing hyperparameters for batch size, learning rate, and number of training epochs from the following ranges: $range_{batch\ size}$={16,32}, $range_{learning\ rate}$={2e-5, 3e-5, 5e-5}, and $range_{epochs}$={1,2,3,4,5}. For the POS, NER, IC, NC, and WNLI tasks, we reuse the same parameters from Lothritz et al. (2022), for the remaining tasks, we perform a grid search using the original LuxemBERT model to find the best-performing configuration of parameters. Table 2 shows the chosen hyperparameters for each task. We finetune each of our models on the same sets of hyperparameters.

### 3.5 Perturbation Techniques

In order to evaluate the robustness of our models, we investigate three perturbation techniques, some of which are described by Ribeiro et al. (2020): sentence negation, entity replacement, and synonym replacement. For this study, we conduct our experiments as follows: we train our models on unperturbed training and validation sets, and then test them on both the unperturbed and the perturbed test sets, allowing us to determine the robustness of our models to each perturbation technique. Due to the nature of our tasks, we cannot apply each perturbation technique to every test set. Table 3 shows an overview of the techniques we use.

#### 3.5.1 Negation

As described in Section 3.3.6, the aim of sentence negation is to turn a given sentence into a negative. By applying sentence negation to the sentiment analysis, we can change the polarity of sentences, turning positive sentences into negative ones and vice versa. Furthermore, we can apply the technique to RTE by negating one sentence of each *entailment* pair in the test set. This approach will turn an *entailment* sentence pair into a *not_entailment* pair.

#### 3.5.2 Entity Replacement

Entity Replacement describes replacing proper names such as person's or location names in the datasets. Intuitively, changing names should not alter the meaning of sentences in our datasets, so the predictions of the models should remain the same regardless of the test set we use. For this study, we focus on replacing first names as well as location names as they are the most common types of names in our datasets. Specifically, we replace names in each sentence in our test sets by a randomly chosen one from the same list of first names that was used to augment the pre-training data for DA BERT (Olariu et al., 2023). In order to maintain consistency, we ensure that identical names in the datasets are all mapped to the same names during the replacement.

#### 3.5.3 Synonym Replacement

As the name implies, for the synonym replacement perturbation, we replace words in the test set by a randomly selected synonym. Specifically, we replace 0 or 1 synonym in each sentence in each of our test sets. Similarly to entity replacement, this kind of perturbation technique should not change the meaning of a given sentence and thus not modify the prediction of a model. For this, we use the same synonym dictionary that was used to augment the pre-training corpus for DA BERT.

[7] www.rtl.lu

[8] We make the dataset available on request

| Task | POS | NER | IC | NC | WNLI | SN | RTE | SA1 | SA2 |
|---|---|---|---|---|---|---|---|---|---|
| batch size | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| learning rate | 5e-5 | 5e-5 | 5e-5 | 2e-5 | 5e-5 | 5e-5 | 5e-5 | 3e-5 | 5e-5 |
| # epochs | 3 | 3 | 5 | 2 | 5 | 4 | 4 | 2 | 2 |

Table 2: Fine-tuning hyperparameters for each investigated task

| PT | POS | NER | IC | NC | WNLI | SN | RTE | SA1 | SA2 |
|---|---|---|---|---|---|---|---|---|---|
| Negation | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Name replacement | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Location replacement | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ |
| Synonym replacement | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |

Table 3: Applicability of the perturbation techniques

## 4 Experimental Results

In this section, we will present the detailed results from our experiments. Table 4 shows the average performance of each model on each task using the original test sets in terms of F1 score. Table 5 displays the performances on original and perturbed test sets of each model fine-tuned on Sentence Negation, RTE, and Sentiment Analysis.

### 4.1 RQ1: Which model yields the highest performance on downstream NLP tasks?

In order to answer this question, we refer to the results shown in both Table 4 and Figure 1. Both the simple mBERT and GottBERT models perform poorly compared to the remaining models, which is to be expected. In addition, the GottBERT models fine-tuned for WNLI, SN, and RTE are all naive classifiers that consistently predict *not_entailment* for the WNLI task, position *3* for the SN task, and *not_entailment* for the RTE task. However, GottBERT does outperform each model in the POS-tagging task, and mBERT outperforms every model except for LB_GottBERT in the WNLI task. On the other hand, both the Lb_mBERT and Lb_GottBERT models almost consistently outperform each remaining model, with Lb_GottBERT performing best in four out of nine tasks, and Lb_mBERT performing best in two tasks and second-best in four tasks. The two models that were pre-trained from scratch usually achieve intermediate performances. However, one notable exception is the SA1 task where both outperform Lb_mBERT and Lb_GottBERT with DA BERT significantly outperforming every other model.

### 4.2 RQ2: How robust are models against data perturbation?

In order to answer this question, we applied the perturbation techniques as described in Section 3.5 to the test sets from three of the investigated tasks: Sentence Negation, RTE, and Sentiment Analysis. For each perturbation technique, we only consider the samples that were affected, omitting the samples that were unchanged during the perturbation process. We then test each fine-tuned model on both the original and the perturbed test sets we generated. We report the differences in performance of each model between the unperturbed and perturbed test sets for SN, RTE, and SA in Table 5.

Overall, we notice that both negation and synonym replacement perturbations have a moderate to high impact on the performance of the models, while name and location replacements have a relatively low impact (cf. Fig. 2, 3, 4, 5)

For the SN task, we notice that both entity perturbation techniques, name replacement and location replacement, generally have a very low impact on the performance of the chosen models. One noticeable outlier is the original LuxemBERT model with an average difference of 1.8 percentage points for name replacement, and 3.7 percentage points for location replacement, showing that fine-tuned LuxemBERT models are somewhat susceptible to this kind of data perturbation. Another outlier is the GottBERT model as there is no difference in performance between the perturbed and unperturbed test sets, but as already mentioned, this particular model always predicts *3*. As such, this score is not meaningful. While the differences are very low for entity replacements, we notice significant differences for synonym replacement, most of which are close to

| Task | mBERT | GottBERT | LuxemBERT | DA BERT | Lb_mBERT | Lb_GottBERT |
|---|---|---|---|---|---|---|
| POS | 0.886 | 0.902 | 0.890 | 0.887 | 0.889 | 0.900 |
| NER | 0.689 | 0.661 | 0.700 | 0.708 | 0.717 | 0.726 |
| IC | 0.460 | 0.574 | 0.725 | 0.717 | 0.760 | 0.762 |
| NC | 0.900 | 0.871 | 0.918 | 0.900 | 0.906 | 0.900 |
| WNLI | 0.640 | 0.780* | 0.596 | 0.544 | 0.560 | 0.650 |
| SN | 0.804 | 0.248* | 0.859 | 0.858 | 0.867 | 0.883 |
| RTE | 0.488 | 0.512* | 0.528 | 0.551 | 0.563 | 0.489 |
| SA1 | 0.612 | 0.636 | 0.666 | 0.687 | 0.664 | 0.651 |
| SA2 | 0.737 | 0.697 | 0.859 | 0.861 | 0.868 | 0.864 |

Table 4: Results for each task on the original test sets. * denotes naive classifier that always predicts the same class

| Perturbation | #samples | mBERT | GottBERT | LuxemBERT | DA BERT | Lb_mBERT | Lb_GottBERT |
|---|---|---|---|---|---|---|---|
| Sentence Negation | | | | | | | |
| NR | 356 | 0.1 | 0.0 | 1.8 | 0.6 | 0.2 | 0.5 |
| LR | 527 | 0.9 | 0.0 | 3.7 | 1.7 | 1.1 | 1.6 |
| SR | 6597 | 13.0 | 0 | 14.2 | 6.9 | 12.7 | 13.8 |
| Recognizing Textual Entailment | | | | | | | |
| Neg | 373 | 100 | 100 | 38.2 | 41.1 | 2.5 | 41.6 |
| NR | 243 | 0 | 0 | 2.3 | 2.4 | 2.4 | 3.4 |
| LR | 363 | 0 | 0 | 2.0 | 3.4 | 0.3 | 5.7 |
| SR | 682 | 0 | 0 | 0.2 | 0.6 | 0.6 | 5.1 |
| Sentiment Analysis 1 | | | | | | | |
| Neg | 45 | 8.7 | 5.1 | 22.1 | 32.3 | 20 | 19.5 |
| NR | 11 | 4.3 | 0 | 1.5 | 4.3 | 0 | 2 |
| LR | 24 | 2.8 | 2.2 | 6.3 | 4 | 3.1 | 3.6 |
| SR | 276 | 0.5 | 0.6 | 0.9 | 0.6 | 1.1 | 1.2 |
| Sentiment Analysis 2 | | | | | | | |
| Neg | 1587 | 19.6 | 24.2 | 27.5 | 33.1 | 36.0 | 33.6 |
| NR | 148 | 0.9 | 1.0 | 1.0 | 1.8 | 0.8 | 1.4 |
| SR | 1508 | 1.1 | 5.3 | 0.9 | 2.6 | 2.2 | 2.0 |

Table 5: Difference (in percentage points) of performances between original test sets and perturbed sets (Neg: Negated test set / NR: Test set with name replacement/ LR: Test Set with location replacement/ SR: Test set with synonym replacement)

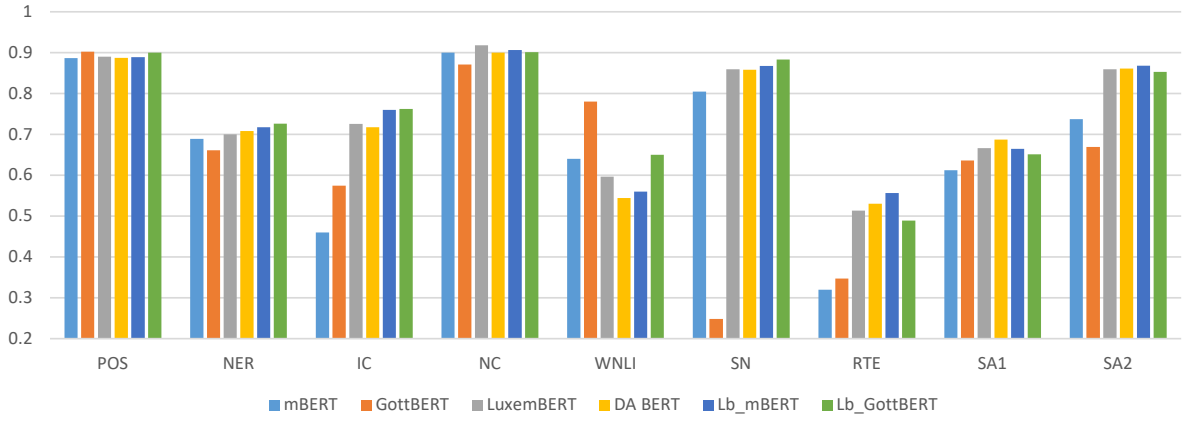


Figure 1: Fine-tuning results of the models on each investigated task

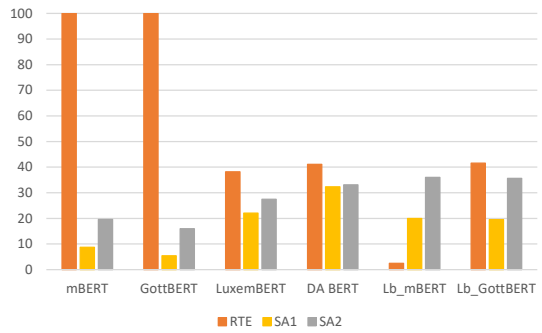


Figure 2: Impact of negation on each model's performance.

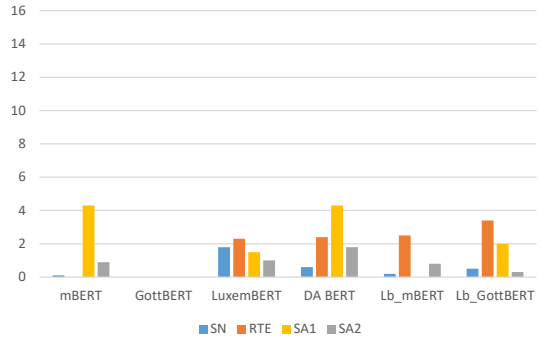


Figure 3: Impact of name replacements on each model's performance.

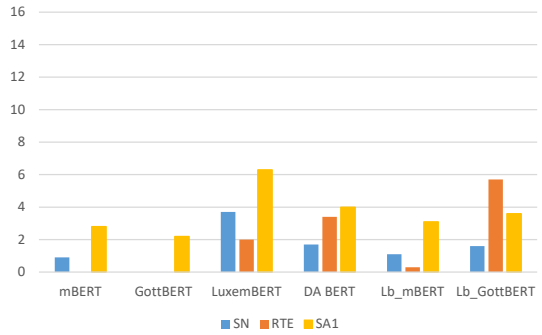


Figure 4: Impact of location replacements on each model's performance.

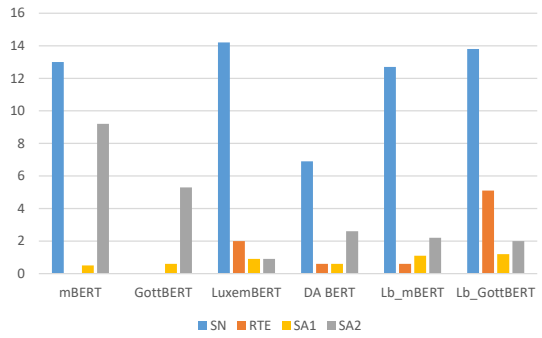


Figure 5: Impact of synonym replacements on each model's performance.

10 percentage points. Once again, the LuxemBERT model shows the highest difference with 14.2 percentage points. DA BERT, which was partially trained on data that was augmented with synonym replacements, shows to be more robust against this kind of data perturbation compared to the remaining models with a difference of only 6.9 percentage points. For the RTE task, we observe that most models with the exception of Lb_GottBERT are fairly robust against the replacement perturbation techniques. On the other hand, they are very susceptible to negation, as only Lb_mBERT's performance is almost unchanged when tested on perturbed data; each remaining model's performance is nearly 40 percentage points lower. We notice a similar trend on the SA2 task, where replacement techniques have only a slight impact on the model performance while negation has a high impact, the difference in performance ranging from nearly 20-35 percentage points depending on the model. Regarding the SA1 task, we observe low, yet mixed results for both entity replacement techniques, but this might be due to the very small sample size of the respective datasets. On the other hand, the impact of sentence negation and synonym replacement is noticeably smaller compared to the SA2 task across all models.

## 5 Discussion

We show that it is possible to achieve higher performance with the same amount of pre-training data and training time as pre-training from scratch, making our approach both more data- and time-efficient. Overall, both Lb_mBERT and Lb_GottBERT outperform LuxemBERT and DA BERT in almost all tasks. (cf. Table 4) However, while Lb_mBERT is also shown to be highly resistant to data perturbation, it appears that the impact of perturbation on Lb_GottBERT's performance varies depending on the task. On the other hand, both models trained from scratch display worse resistance to data perturbation than Lb_mBERT. As such, we conclude that it is preferable to continue pre-training a pre-existing model on textual data in the target language. According to our experiments, it appears that there is a trade-off between performance and robustness depending on the choice of pre-trained language model. A multilingual model should be chosen if robustness is preferred, while a model for a language that is close to the target language is preferable if the objective is high performance, at

least judging by the results from our experiments.

## 6 Related Works

Wu and Dredze (2020) proposed pairing related languages to train a low-resource language model can result in a performance improvement over a monolingual model. In particular, they combined Latvian and Lithuanian text to create a Latvian BERT model as well as Afrikaans and Dutch text to create an Afrikaans BERT model. Similarly, the Luxembourgish LuxemBERT model (Lothritz et al., 2022) was also trained on bilingual data joining Luxembourgish and German text. However, while those language models are jointly pre-trained on data written in different languages from scratch, for our approach, we pre-train already existing language models on new language data.

Similar to our approach, Muller et al. (2021) continued to pre-train mBERT to various unseen low-resource languages written in different non-Latin scripts and evaluate the performance on three common NLP tasks. Similar to our own experiments, they found that this approach typically leads to models that outperform both the original mBERT and models that were trained from scratch. Our study, however, focuses on a single language that is featured in mBERT. Furthermore, we do not only apply this approach to mBERT, but also to GottBERT to evaluate the performance gain of pre-training a pre-loaded model for a language that is close to the target language.

Ribeiro et al. (2020) introduced CheckList, a tool to semi-automatically create a large number of test cases to determine the robustness of NLP models. Similarly to our study, they consider various types of simple data perturbations to create new test samples. However, their tool is more versatile as it also allows the creation of templates to generate a large number of simple sentences as well as simple additions of phrases that do not change the label of a sample.

## 7 Threats to Validity

Similar to most experimental studies, there are factors that might threaten the validity of this work when scrutinised.

The first threat is related to the choice of the pre-loaded models, namely mBERT and GottBERT. Both of these models were pre-trained with hyperparameters that slightly differ from the LuxemBERT and DA BERT models, so the improved performance might have been due to confounding variables that we did not control. In particular, the alphabet size and vocabulary size differ significantly as mentioned in Section 2.1. However, we deemed GottBERT and mBERT as appropriate baselines for our study as they are the closest to LuxemBERT and DA BERT in terms of architecture.

Another possible threat concerns some of the downstream tasks we chose to evaluate our models. Specifically, the RTE and SA1 tasks are problematic as they were automatically translated without manually correcting the result. As such, there are numerous translation mistakes present in these datasets which might have influenced the results of our experiments.

## 8 Conclusion

In this study, we investigated the effects of pre-training pre-loaded language models vs pre-training language models from scratch for building Luxembourgish language models. We evaluated our models in two dimensions: performance and robustness. We conducted our experiments on nine downstream NLP tasks of varying difficulty, and invesitgated the robustness of our models with three perturbation techniques. We found that pre-training a pre-loaded model does indeed have a positive effect on both the performance and robustness of fine-tuned models. In particular, the results from our experiments suggest that using the German GottBERT model yields a higher performance, while the multilingual mBERT results in a more robust model.

## 9 Acknowledgements

This work was partially supported by the Fonds National de la Recherche (FNR), Luxembourg, under project LUXEMBERT BRIDGES2021/IS/16229163.

## 10 Limitations

The approach presented in this work was only tested on the Luxembourgish language and using German as an auxiliary language. The approach should be generalisable to other languages, but this might be limited by how similar the auxiliary language is to the target language in terms of structure and vocabulary. We are confident that the approach for continued pre-training is applicable if the target language is either a dialect of or part of the

same language family as the language of the pre-loaded language model. However, the applicability of this approach is unclear for languages that differ significantly from each other.

## 11 Ethical Considerations

This study involved a pre-training corpus that partially consists of user comments from a news website and chatlogs from a defunct chatroom, both of which originally included usernames (Lothritz et al., 2022). However, this data was anonymised before model training. While we do publish our models that were trained with the same data, we do not publish the pre-training corpus in question. The remaining datasets that we publish are all based on either publicly available textual data dumps or already existing datasets from the GLUE collection, and as such do not violate GDPR guidelines to the best of our knowledge.

## References


Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *2019 Conference of the North American Chapter of the ACL: Human Language Technologies*.

Peter Gilles. 2022. Luxembourgish. In *The Oxford Encyclopedia of Germanic Linguistics*.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7.

Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2020. A survey on recent approaches for natural language processing in low-resource scenarios. *arXiv preprint arXiv:2010.12309*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Cedric Lothritz, Kevin Allix, Bertrand Lebichot, Lisa Veiber, Tegawendé F Bissyandé, and Jacques Klein. 2021. Comparing multilingual and multiple monolingual models for intent classification and slot filling. In *International Conference on Applications of Natural Language to Information Systems*, pages 367–375. Springer.

Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Lisa Veiber, Tegawendé Bissyandé, Jacques Klein, Andrey Boytsov, Clément Lefebvre, and Anne Goujon. 2022. Luxembert: Simple and practical data augmentation in language model pre-training for luxembourgish. In *Proceedings of the Language Resources and Evaluation Conference*, pages 5080–5089, Marseille, France. European Language Resources Association.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mbert is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462.

Isabella Olariu, Cedric Lothritz, Tegawendé F. Bissyandé, and Jacques Klein. 2023. Evaluating Data Augmentation Techniques for the Training of Luxembourgish Language Models. In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, and Martin Boeker. 2020. Gottbert: a pure german language model. *arXiv e-prints*, pages arXiv–2012.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. 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*, pages 353–355.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert? In *5th Workshop on Representation Learning for NLP*, pages 120–130.