LuxemBERT: Simple and Practical Data Augmentation in Language Model Pre-Training for Luxembourgish

Cedric Lothritz[1], Bertrand Lebichot[1], Kevin Allix[1], Lisa Veiber[1], Tegawendé F. Bissyandé[1], Jacques Klein[1], Andrey Boytsov[2], Anne Goujon[2], Clément Lefebvre[2]

[1]University of Luxembourg

[1]6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg

[2] Banque BGL BNP Paribas

[2] 50, avenue J.F Kennedy, L-2951 Luxembourg

{cedric.lothritz, bertrand.lebichot, kevin.allix, lisa.veiber, tegawende.bissyande, jacques.klein}@uni.lu

{andrey.boytsov, anne.goujon, clement.c.lefebvre}@bgl.lu

Abstract

Pre-trained Language Models such as BERT have become ubiquitous in NLP where they have achieved state-of-the-art performance in most NLP tasks. While these models are readily available for English and other widely spoken languages, they remain scarce for low-resource languages such as Luxembourgish. In this paper, we present LuxemBERT, a BERT model for the Luxembourgish language that we create using the following approach: we augment the pre-training dataset by considering text data from a closely related language that we partially translate using a simple and straightforward method. We are then able to produce the LuxemBERT model, which we show to be effective for various NLP tasks: it outperforms a simple baseline built with the available Luxembourgish text data as well the multilingual mBERT model, which is currently the only option for transformer-based language models in Luxembourgish. Furthermore, we present datasets for various downstream NLP tasks that we created for this study and will make available to researchers on request.

Keywords: Language Models, Less-Resourced Languages, NLP Datasets

1. Introduction

Pre-trained Language Models for NLP tasks have become increasingly popular over the last years and will likely continue to thrive in the years to come. Their usefulness is immediately obvious as they mitigate the need to train specific NLP models from scratch and can be reused for multiple tasks through fine-tuning. In particular, BERT (Devlin et al., 2018) and its variants such as RoBERTa (Liu et al., 2019), DistilBERT (Sanh et al., 2019), and XLNet (Yang et al., 2019) are widely leveraged by researchers and practitioners alike. Unfortunately, while these models generally reach state-of-the-art performances for most downstream tasks, they present a significant caveat as the pre-training step requires huge amounts of computing resources, time, and, most importantly, data. For example, the original BERT model, which targeted the English language, was trained on the entire English version of Wikipedia as well as the BooksCorpus (Zhu et al., 2015), amounting to a 3.3 billion words dataset. While this amount of data is readily available for widely spoken languages such as English, German and French, it is not the case for many low-resource languages such as Luxembourgish. This data scarcity therefore becomes a major obstacle for building adequate language models.

Data from low-resource languages have been included along many other languages to build a multilingual BERT (mBERT) which researchers and practitioners resort to for dealing with NLP tasks. Unfortunately, although mBERT-based models generally perform well, they are usually outperformed by monolingual models if an adequate amount of data is available (Wu and Dredze, 2020). To get enough data, Wu and Dredze (2020) have recently proposed to augment pre-training datasets by adding textual data from a different language that is closely related to the target language. We explore this direction in our research. In this paper, we introduce LuxemBERT, a BERT-like model for Luxembourgish[1], a West Germanic language that is closely related to German. In order to overcome the challenge of data scarcity, we propose an approach focusing on improving the suitability of the textual data collected from an auxiliary language. We propose to partially translate a subset of widely common and unambiguous words from the auxiliary language to the target language, in order to make the supplementary dataset resemble more closely the limited dataset of the target language. Using this approach, we combine Luxembourgish and German data to build an adequate pre-training corpus to build LuxemBERT. To assess the effectiveness of LuxemBERT, we build several datasets for a variety of downstream NLP tasks. We compare its performance to the de facto state of the art based on mBERT as well as to a baseline built by training a BERT model with the limited text data available in Luxembourgish. Our contributions are threefold:

(a) LuxemBERT, a cased and uncased BERT model for the Luxembourgish language,

[1]one of the official languages of Luxembourg.

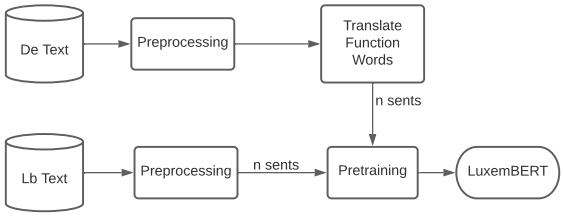


Figure 1: Data augmentation scheme for LuxemBERT (De: German / Lb: Luxembourgish)

(b) Annotated datasets for four NLP-tasks to evaluate Luxembourgish language models that we make available to the research community,

(c) A strategy to augment pre-training data for low-resource languages.

The rest of this paper is structured as follows: In Section 2, we present our LuxemBERT model, the pre-training dataset we use, and the training hyperparameters. In Section 3, we present our baseline models, the datasets for the downstream tasks to evaluate LuxemBERT, and the fine-tuning parameters. In Section 4, we present the results of our experiments, address the research questions, and report the performance of LuxemBERT. Section 5 discusses the results we obtained. In Section 6, we present some potential threats to the validity of our study. Section 7 discusses a selection of works related to this paper. Finally, we conclude our findings in Section 8.

2. LuxemBERT

Wu and Dredze (2020) proposed to pair two closely related languages to increase the quality of the learned embeddings. Inspired by this approach, we aim to create a novel augmented dataset. However, we seek to decrease the differences between the dataset written in the auxiliary language and the one written in the target language. To this end, we partially and systematically translate common and unambiguous words into the target language. Intuitively, we expect this approach to decrease noise introduced by the auxiliary language and further improve the learned word embeddings. Bernhard and Ligozat (2013) proposed a similar method for Part-of-Speech (POS) tagging where they systematically translate a selection of words from Alsatian sentences to German and evaluate the performance of a German POS-tagger on the resulting dataset.Using our approach, we train a BERT model for the Luxembourgish language, which we appropriately name LuxemBERT.[2] Figure 1 shows the pre-training schema of our LuxemBERT model. For the creation of the pre-training corpus, we take advantage of the similarity between Luxembourgish and German. There is a sizeable overlap between the vocabularies between both

Meaning (for readers) (English)	*There are 26 known isotopes, only two of which appear in nature.*
Sample text in auxilliary language (German)	Bekannt sind 26 Isotope, wovon nur zwei natürlich vorkommen.
Translated text for data augmentation (pseudo-Luxembourgish)	Bekannt sinn 26 Isotope, wouvun nëmmen zwee natierlech vorkommen
Ground-truth translation (Luxembourgish)	*Bekannt sinn 26 Isotopen, wouvun der nëmmen zwee natierlech virkommen*

Figure 2: Example pseudo-translation for LuxemBERT

languages. Indeed, we downloaded a list of 19 366 Luxembourgish-German word pairs[3], and determined that 3809 word pairs are identical, 3489 word pairs have a Levenshtein distance of 1 and 2333 pairs have a distance of 2. Furthermore, as both languages are closely related from a structural standpoint, it is possible to translate single words from one language to the other without significantly changing the meaning of the sentence. We exploit this feature to build a simple mapping table to partially translate the German portion of the pre-training corpus to Luxembourgish. Specifically, we translate unambiguous function words. Function words are usually defined as words that have little to no meaning on their own, but are mainly used to structure a sentence (Carnie, 2021). Examples for function words include determinants, pronouns, prepositions, and numerals. In contrast to content words such as nouns, verbs, or adjectives, function words are few in number, but make up a sizeable portion of everyday texts, allowing to translate a sizeable portion of the text with relatively little effort. Indeed, Pennebaker (2011) suggests that the English language contains around 450 function words which, in spite of the small number, make up 55 percent of the words people use. Due to these properties, we deem function words appropriate candidates for the translation strategy. We identified a list of 529 unambiguous German/Luxembourgish function word pairs. Using a mapping table, we automatically translate a portion of the German part of our pre-training dataset. Specifically, this method allows us to translate nearly 20% of the German part of the dataset. Figure 2 shows an example sentence that was translated using our mapping table. Note that this pseudo-translation is nearly identical to the actual translation despite the simplicity of the method.

In order to determine the appropriate amount of augmented data to add to the dataset, we created several datasets containing half a million sentences each, varying the ratio of Luxembourgish and German data for every set. The datasets contain 0%, 20%, 40%, 50%, 60% 80%, and 100% German data, respectively. We then fine-tuned each resulting model on five downstream tasks over five runs, and averaged the performances. Figure 3 shows the results of our experiment. While we find that the model pre-trained on 100% German data usually performs worst, the performances of

[2]The final (uncased) model can be found at `https://huggingface.co/lothritz/LuxemBERT`

[3]`https://github.com/robertoentringer/appli`

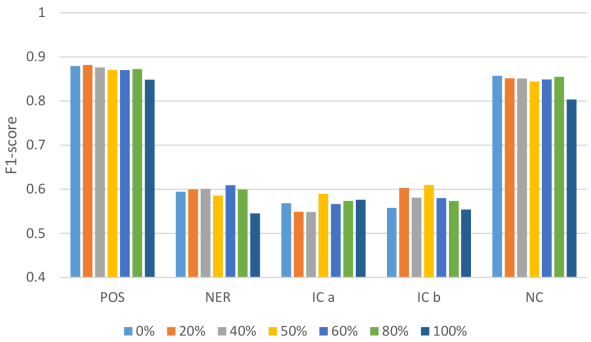


Figure 3: Results of experiments for determining best Lb/De ratio for LuxemBERT

the remaining models are mixed. However, we find that the model trained on 50% Luxembourgish and 50% German data achieved the highest mean and lowest standard deviation across all tasks. Following this result, we pre-train LuxemBERT on 50% Luxembourgish and 50% translated German data.

2.1. Dataset for Pre-training

We collected textual data from various sources such as news articles and the Luxembourgish version of Wikipedia. In total, we collected nearly 6.1 million sentences written in Luxembourgish. Table 1 shows a breakdown of the used corpus. In order to assess the impact of the corpus size on the performance of the model, we trained models with three different subsets of the corpus (*small*, *medium*, and *large*). The *small* dataset consists of the entirety of the Luxembourgish Wikipedia only. Specifically, we downloaded the most recent version on March 10, 2021, with wp-download[4], making up nearly 500 000 sentences. The *medium* dataset consists of the Luxembourgish Wikipedia, as well as news articles and webpages featured in the Leipzig Corpora Collection (Goldhahn et al., 2012). Specifically, we downloaded 300 000 sentences of the Newscrawl dataset, 1 million sentences of the 2013 Web dataset, and 300 000 sentences of the 2015 Web dataset. In total, this dataset consists of 2.1 million sentences. Finally, the *large* dataset contains each of the aforementioned sets, as well as news articles, radio broadcast transcripts, and pseudonymised user comments from the Luxembourgish News station RTL.[5] In addition, it contains pseudonymised chatlogs from the defunct Luxembourgish Chatroom Luxusbuerg and example sentences from the Luxembourgish Online Dictionary.[6] We are aware of the OSCAR dataset (Ortiz Suárez et al., 2020) which contains Luxembourgish text, however, it is mostly made up of Wikipedia articles which would result in a large number of duplicate sentences in our dataset. As such, we omit the dataset for pretraining.

source	#sentences
Wikipedia	500k
News articles	300k
Webpages	1.3M
RTL user comments	1.57M
RTL news articles	1.64M
RTL radio broadcasts	572k
Chatroom logs	175k
LOD	50k
Total	6.1M

Table 1: Breakdown of pre-training corpus

In total, the data amounts to more or less 6.1 million sentences or 130 million words, a sizeable difference to the corpus used to train the original BERT model by Devlin et al. (2018) which consists of 3.3 billion words. For the German part of the dataset, we collected articles from the German Wikipedia, for an additional 6.1 million sentences.

2.1.1. Training Parameters

The *BERT Base* model created by Devlin et al. (2018) contains 12 transformer blocks, 768 hidden layers, 12 self-attention blocks, and 110 million parameters in total. We reuse the same configuration to pre-train LuxemBERT. However, in contrast to the original BERT model, we drastically reduce the alphabet size from 1000 to 120 to accommodate the Luxembourgish alphabet. The pre-training is done using the Masked Language Modeling task over 10 epochs and with masking probability of 15%. The sentences in our pre-training corpus were largely unordered, making it difficult to build an adequate dataset for the Next Sentence Prediction task, which is why we omitted that task from the pre-training step. The pre-training was done using the HPC facility at the University of Luxembourg (Varrette et al., 2014).

3. Experimental Setup

In this section, we enumerate the research questions, describe the baselines to compare against LuxemBERT, and discuss the downstream tasks on which the models are assessed.

3.1. Research questions

We investigate the following research questions:

- RQ1: *Does LuxemBERT outperform the state of the art for Luxembourgish-targeted NLP tasks?* We consider multi-lingual BERT (mBERT) as the main comparison point to demonstrate the added-value of LuxemBERT on several tasks.
- RQ2: *Is our data augmentation scheme effective for improving model pre-training?* We assess the effectiveness of our approach by proposing an ablation study where we compare LuxemBERT against a BERT model trained with available Luxembourgish

[4] https://github.com/pacurromon/wp-download

[5] www.rtl.lu

[6] www.lod.lu

Task	train	dev	test
POS	4291	459	750
NER	4291	459	750
IC a	698	149	159
IC b	606	130	137
NC	7057	1034	1961
WNLI	568	63	136

Table 2: Breakdown of datasets used for fine-tuning LuxemBERT on downstream tasks

Task	#labels	max	min	mean	median
POS	15	16452	7	4864	3915
NER	5	2272	95	1214	1239
IC a	28	60	23	36	35.5
IC b	23	60	28	38	37
NC	8	2866	106	1257	1120
WNLI	2	409	358	383.5	383.5

Table 3: Statistics for labels of datasets used for fine-tuning LuxemBERT on downstream tasks

text data. We further evaluate the impact of our partial translation scheme by comparing LuxemBERT against a version where the augmented dataset is non-translated German.

3.2. Baselines

We consider three baselines for comparison: multilingual BERT; a pure Luxembourgish BERT; and a Bilingual BERT (trained with Luxembourgish and German data).

3.2.1. mBERT

There is currently no existing transformer-based model for the Luxembourgish language. Therefore, we use the multilingual version of BERT[7] as a baseline to evaluate the performance of the LuxemBERT models on the selected downstream tasks. mBERT has been trained on Wikipedia articles in more than 100 languages, including Luxembourgish. mBERT contains 12 transformer blocks, 768 hidden layers, 12 self-attention blocks, and 110 million trainable parameters, and was released as a cased and an uncased version. The Luxembourgish component of mBERT was trained using the entire Luxembourgish Wikipedia, which consisted of 59 000 articles at the point of training.

3.2.2. Lb BERT: Simple Luxembourgish BERT

As a second baseline, we use a BERT model that we pre-train on Luxembourgish data only. This allows us to determine the impact of adding augmented data on the performance of the language model. This baseline will be called Lb BERT.

3.2.3. Lb/De BERT: Bilingual BERT

Following the approach by Wu and Dredze (2020), we train a bilingual BERT model as our final baseline. Similarly to LuxemBERT, the dataset for this model consists of 50% Luxembourgish and 50% German data. It will be referred to as Lb/De BERT.

3.3. Downstream Tasks

We consider five down-stream tasks to assess the performance of our LuxemBERT model: Part-of-Speech (POS) tagging, Named Entity Recognition (NER), Intent Classification (IC), News Classification (NC) and the Winograd Natural Language Inference (WNLI) task. As suitable datasets are scarce, we created a number of Luxembourgish ones ourselves. Table 2 shows an overview of each dataset used for fine-tuning and Table 3 shows various statistics for the labels of each downstream dataset. As most of these datasets are based on articles from RTL, we cannot publish them, but we make them available to researchers on request.

3.3.1. Part-of-Speech Tagging

Part-of-Speech tagging is a fundamental sequence-to-sequence task, the objective of which consists of assigning a grammatical class such as *noun*, *verb*, or *adjective* to each word in a given sentence. For this dataset, we downloaded several months worth of written news articles from RTL which cover topics such as politics, local and world news, sports, and tabloid news. We made sure not to reuse data from the pre-training corpus. The dataset consists of 450 Luxembourgish news articles, totalling 5500 sentences. We consider 15 typical POS-tags. The tagging for this groundtruth dataset was done using a Luxembourgish spaCy model[8] and verified by a native Luxembourgish speaker. The biggest class is the *Noun* class with 16 452 samples, the smallest is the *Interjection* class with 7 samples, the mean sample count per class is 4864 while the median is 3915.

3.3.2. Named Entity Recognition

Similarly to POS-tagging, Named Entity Recognition is a sequence-to-sequence task. The objective is to detect and differentiate between proper names such as persons, locations, or organisations. We use the same dataset that we use for POS-tagging, i.e., a collection of news articles downloaded from RTL. We consider five labels: *Person*, *Organisation*, *(natural) Location*, *Geopolitical Entity*, and *Miscellaneous*. As there is currently no NER-tagger available to the best of our knowledge, the set was annotated manually by a single native speaker. The dataset consists of 450 news articles, amounting to 5500 sentences. There is a total of 107 521 words, 101 453 of which are non-entities, and 6068 are named entities. The *Person* class is the biggest with 2272 samples, *Location* is the smallest with 95 samples, the mean is 1214, and the median is 1239.

[7] https://github.com/google-research/bert/blob/master/multilingual.md

[8] https://github.com/PeterGilles/Luxembourgish-language-resources/blob/master/spaCy%20for%20Luxembourgish.ipynb

3.3.3. Intent Classification

Intent Classification consists of inferring the author's intention based on a piece of text. For this task, we use the Banking Client Support dataset created by Lothritz et al. (2021). It contains 1006 samples divided into 28 different intents related to banking requests such as opening/closing a bank account or ordering/blocking a credit card. The biggest class is *check_balance* with 60 samples while the smallest class is *goodbye* with 23 samples. The average samples count per class is 36 while the median is 35.5. We split this dataset into two subsets: (a) the entire dataset as is, (b) a set containing only the 'non-trivial' intents, with the following intents removed from the original dataset: *affirm*, *deny*, *greet*, *goodbye*, and *thankyou*. This subset contains 863 samples divided into 23 intents. The biggest class is again the *check_balance* class with 60 samples, and the smallest is *check_recipients* with 28 samples. The average sample count is 38 and the median is 37.

3.3.4. News Classification

News Classification is a common NLP-task, consisting of categorising given news articles into topics such as *sports*, *political*, or *tabloid* news. We scraped news articles from RTL and selected a variety of topics, ensuring that there is no overlap with the data we used for pre-training. Specifically, we chose *national*, *European*, and *global* news, as well as articles about *sports*, *culture*, *gaming*, *technology*, and *cooking recipes*, for a total of 8 categories. The annotating was done using the metadata of the article pages. The dataset contains 10 052 articles. The *sports* class is the biggest with 2866 articles while there are merely 106 *recipes* articles. On average, there are 1257 articles per class, and the median is 1120.

3.3.5. Winograd Natural Language Inference

The Winograd Natural Language Inference (WNLI) task is part of the GLUE benchmark collection (Wang et al., 2018). The dataset consists of text pairs, where text A contains one or several pronouns. Text B consists of a substring of text A where a pronoun is replaced by a word or a name. The label is either 1 or 0 depending on whether or not the pronoun was replaced with the correct token from text A. The WNLI dataset was originally created by Levesque et al. (2012). We translated the dataset to Luxembourgish.[9] Furthermore, as the labels for the test set are not public, we annotated it ourselves. The final dataset contains 767 samples. There are 409 samples with the *0* label and 358 with the *1* label.

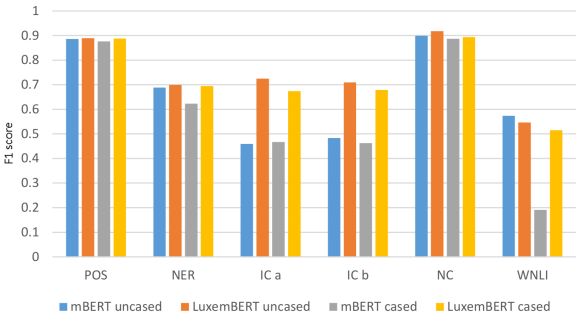
3.4. Fine-tuning Parameters

Regarding fine-tuning parameters, Devlin et al. (2018) report that the best performances for downstream NLP tasks are observed for a batch size in $\{16, 32\}$, a learning rate in $\{2^{-5}, 3^{-5}, 5^{-5}\}$, and training epochs in $\{2, 3, 4\}$. We perform a grid search to determine which of these parameters yield the highest performance when fine-tuning an uncased mBERT model, and use these parameters for the remaining models. The parameters for every downstream task are given in Table 4.

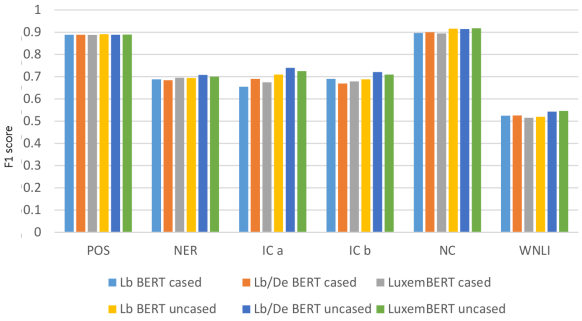
Task	batch size	learning rate	#epochs
POS	16	5^{-5}	3
NER	16	5^{-5}	3
IC a	16	5^{-5}	5
IC b	16	5^{-5}	5
NC	16	2^{-5}	2
WNLI	16	5^{-5}	5

Table 4: Results of grid search for parameters

4. Experimental Results



(a) mBERT vs LuxemBERT



(b) Lb BERT vs Lb/De BERT vs LuxemBERT

Figure 4: Comparison of mBERT, Lb BERT, Lb/De BERT, LuxemBERT on the large data set

In this section, we present and analyse the results of our experiments and answer the research questions we asked in Section 3.1. As mentioned in Section 2.1, we pre-train our models with three dataset sizes named *small*, *middle*, and *large*. Furthermore, we train both *cased* and *uncased* models for every given dataset. In order to evaluate the performance of our BERT models, we separately fine-tune the pre-trained models on each downstream task over five runs, resulting in five fine-tuned models per task and per pre-trained model. We then calculate the average performance of each fine-tuned model in terms of F1-score. Tables 5 and 6 show

[9] The final dataset can be found at `https://github.com/Trustworthy-Software/LuxemBERT-datasets`

Model	POS	NER	IC a	IC b	NC	WNLI
Lb BERT *small*	88.0 ± 0.1	59.4 ± 1.0	56.9 ± 5.3	55.8 ± 4.0	85.7 ± 0.2	51.8 ± 2.1
Lb/De BERT *small*	88.3 ± 0.1	61.5 ± 0.3	54.4 ± 1.7	59.7 ± 2.2	86.9 ± 0.3	49.9 ± 0.0
LuxemBERT *small*	88.0 ± 0.2	61.9 ± 0.5	55.9 ± 2.6	60.1 ± 2.7	87.0 ± 0.3	49.9 ± 0.0
Lb BERT *medium*	88.3 ± 0.1	65.4 ± 0.5	63.4 ± 1.8	63.6 ± 0.8	89.4 ± 0.2	51.7 ± 2.5
Lb/De BERT *medium*	**89.1 ± 0.2**	68.9 ± 0.7	64.4 ± 2.0	67.0 ± 1.8	89.9 ± 0.3	52.2 ± 1.9
LuxemBERT *medium*	88.7 ± 0.2	66.8 ± 0.8	66.2 ± 1.6	69.3 ± 1.1	90.3 ± 0.2	50.8 ± 1.4
Lb BERT *large*	**89.1 ± 0.3**	69.4 ± 1.0	71.0 ± 1.7	68.8 ± 1.2	91.6 ± 0.2	52.0 ± 2.3
Lb/De BERT *large*	88.8 ± 0.1	**70.8 ± 0.8**	**74.0 ± 2.2**	**72.1 ± 1.4**	91.4 ± 0.2	54.3 ± 1.9
LuxemBERT *large*	89.0 ± 0.1	70.0 ± 0.8	72.5 ± 1.1	70.9 ± 1.8	**91.8 ± 0.2**	54.6 ± 1.6
mBERT	88.6 ± 0.1	68.9 ± 1.0	46.0 ± 5.6	48.3 ± 9.4	90.0 ± 0.5	**57.3 ± 0.0**

Table 5: Comparison of results for uncased models on downstream tasks

Model	POS	NER	IC a	IC b	NC	WNLI
Lb BERT *small*	86.6 ± 0.2	54.4 ± 0.6	57.7 ± 3.8	60.5 ± 3.2	84.4 ± 0.5	49.9 ± 0
Lb/De BERT *small*	87.4 ± 0.2	59.3 ± 0.6	59.9 ± 1.9	60.1 ± 1.6	85.1 ± 0.3	49.9 ± 0
LuxemBERT *small*	87.0 ± 0.1	58.8 ± 0.8	59.6 ± 2.9	60.9 ± 0.6	85.2 ± 0.3	51.6 ± 2.0
Lb BERT *medium*	88.6 ± 0.2	62.7 ± 0.7	65.0 ± 2.1	64.1 ± 1.4	87.6 ± 0.2	49.9 ± 0
Lb/De BERT *medium*	88.9 ± 0.1	66.3 ± 0.3	65.5 ± 3.5	68.3 ± 1.1	88.2 ± 0.1	50.8 ± 1.6
LuxemBERT *medium*	**89.0 ± 0.1**	66.5 ± 0.4	65.7 ± 2.1	66.3 ± 2.6	88.9 ± 0.3	50.7 ± 1.6
Lb BERT *large*	88.8 ± 0.1	68.9 ± 0.8	65.5 ± 2.4	**69.0 ± 2.4**	89.6 ± 0.2	**52.5 ± 0.5**
Lb/De BERT *large*	88.9 ± 0.1	68.4 ± 0.2	**69.0 ± 2.6**	66.9 ± 2.9	**90.0 ± 0.1**	**52.5 ± 3.9**
LuxemBERT *large*	88.8 ± 0.1	**69.5 ± 0.5**	67.4 ± 1.9	67.9 ± 2.9	89.4 ± 0.3	51.5 ± 1.8
mBERT	87.6 ± 0.2	62.3 ± 0.4	46.7 ± 4.1	46.3 ± 8.9	88.7 ± 0.5	19.1 ± 0

Table 6: Comparison of results for cased models on downstream tasks

the results (and standard deviation) for the *uncased* and *cased* models, respectively. We notice that generally, the performance of the models increases and the standard deviation decreases as the size of pre-training data increases. It is also to note that for mBERT, we observe a high standard deviation for many of the downstream tasks when compared to the LuxemBERT models.

Comparing all these results can be tedious. To help us, we used two statistical tests: **The Friedman/Nemenyi (F/N)** test (Demšar, 2006). This test is not very powerful (Cohen, 2013) but allows to compare all pairs of models directly and has an easy-to-interpret visualization. The test first computes the rank of each considered approach for all datasets. Then, the plot reports the mean rank R (the higher, the better) for each approach. An approach a is considered as significantly better than another (b) if its mean rank R_a exceeds R_b by critical difference CD, i.e. $R_a > R_b + CD$. **The Wilcoxon test** (Demšar, 2006) compares the difference of performance for a pair of approaches across datasets. It is more powerful than F/N tests because it only considers two alternatives.

4.1. RQ1: Does LuxemBERT outperform the state of the art for Luxembourgish-targeted NLP tasks?

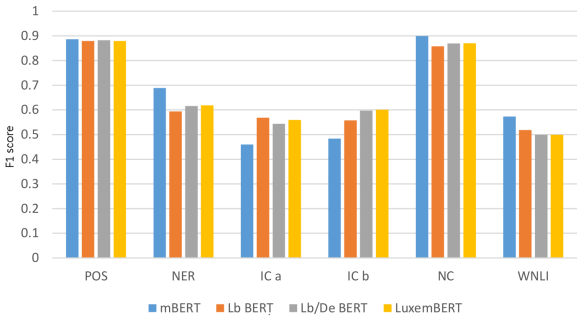
Figure 4a shows a comparison of both mBERT and LuxemBERT models. With regards to the *uncased* models, there is a slight increase in F1-scores for the POS, NER, and NC tasks, and a large increase for IC a, and IC b. On the other hand, the only task on which mBERT outperforms LuxemBERT is the WNLI task. With regards to the *cased* models, LuxemBERT outperforms mBERT on every task, with a slight increase in performance on the POS and NC tasks and a large increase on NER, IC a, IC b, and WNLI.

We perform a Wilcoxon test for LuxemBERT (*small* cased, *medium* cased, *large* cased, *small* uncased, *medium* uncased, and *large* uncased) versus the corresponding mBERT model (cased or uncased). For cased, we find a p-value of 0.219, 0.016, 0.016 for *small*, *medium*, and *large*, respectively. For uncased, we find a p-value of 0.5, 0.281, 0.109 (same order).

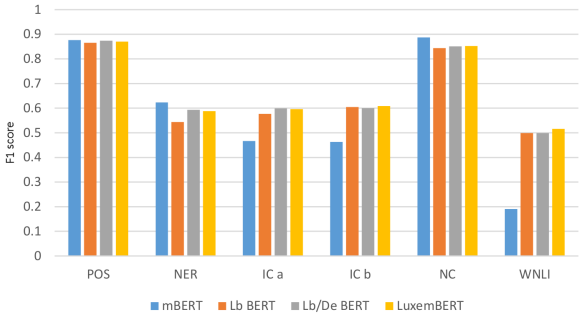
> **RQ1 Answer:** For cased, LuxemBERT outperforms mBERT, even if we train LuxemBERT on a fraction of the data at our disposal. For uncased, LuxemBERT does outperform mBERT, but we needed all data at our disposal.

4.2. RQ2: Is our data augmentation scheme effective for improving model pre-training?

With this second research question, we now want to quantify how Lb/De BERT and LuxemBERT can improve performance by leveraging German data. We compare them to Lb BERT and mBERT. In addition, we leverage the size of the pre-training corpus to quantify how much adding the auxiliary language can improve a language model in the case where the lack of data is even more apparent. Figures 5, 6, and 4b show the performances of our models trained on *small*, *medium* and *large* datasets, respectively. The results of the F/N test can be found in Figures 7b to 7e. From these figures, Lb/De BERT and LuxemBERT clearly emerge as better alternatives, except for *small* (cased and uncased). Lb/De BERT and LuxemBERT are often ahead in terms of performance, with two excep-

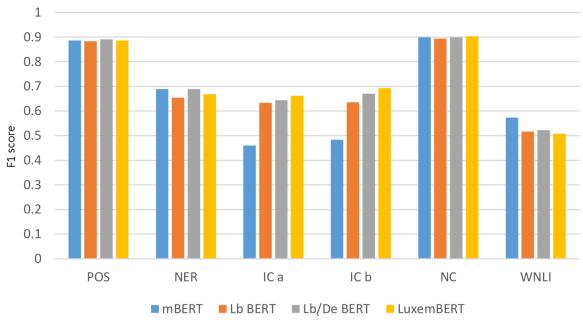


(a) mBERT (uncased) vs LuxemBERT (uncased)

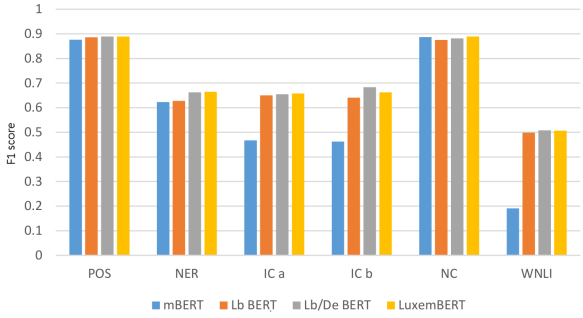


(b) mBERT (cased) vs LuxemBERT (cased)

Figure 5: Comparison of Lb BERT, Lb/De BERT, and LuxemBERT to mBERT on the small-sized data set



(a) mBERT (uncased) vs LuxemBERT (uncased)



(b) mBERT (cased) vs LuxemBERT (cased)

Figure 6: Comparison of Lb BERT, Lb/De BERT, and LuxemBERT to mBERT on the medium-sized data set

tions: (1) for *small* uncased, mBERT seems to be more competitive, and (2) for *large*, Lb BERT is in-between Lb/De BERT and LuxemBERT.

From a statistical point of view, we can learn more by running additional Wilcoxon tests (with p-value=0.05). For cased models, Lb/De BERT and LuxemBERT are superior to Lb BERT for *small* and *medium*. They are also superior to mBERT for *medium* and *large*. For uncased models, Lb/De BERT and LuxemBERT are superior to Lb BERT for *medium*. They are also superior to mBERT for *large*, but only with a p-value around 10%.

RQ2 Answer: The data augmentation strategies of Lb/De BERT and LuxemBERT clearly improve the performance against our baselines. It was not possible to show a statistical difference between both, but LuxemBERT obtained overall better results than Lb/De BERT.

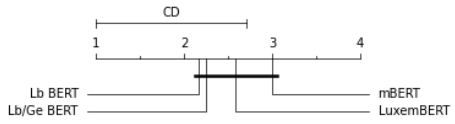
5. Discussion

The main factor of success is the training data size. The second factor is data augmentation: we show that it significantly increases the results among the considered tasks. Finally, we showed that automatic translation can further increase the results.

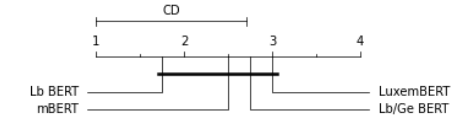
As a last consideration, we compare all variants to search for the best among all 20 alternatives presented in this paper. To do so, we generate the results for all possible pairs of Wilcoxon superiority tests. We assume an alternative is better if the p-value of the superiority test (accounting for the six downstream tasks) is lower than 0.05, as before. We report these results in a Wins/Ties/Losses chart in Table 7. It means that we counted the number of times each of the alternatives significantly beats/was beaten by all 19 others (wins and losses, respectively). When the test cannot conclude because of a large p-value, we call it a tie. The results show that LuxemBERT *large* uncased is the best alternative, and we recommend its usage for NLP in Luxembourgish.

Model name	W	T	L
Lb BERT *small* cased	1	4	14
Lb/De BERT *small* cased	2	1	12
LuxemBERT *small* cased	2	5	12
Lb BERT *medium* cased	6	3	10
Lb/De BERT *medium* cased	9	3	7
LuxemBERT *medium* cased	9	3	7
Lb BERT *large* cased	11	6	2
Lb/De BERT *large* cased	11	5	3
LuxemBERT *large* cased	10	6	3
Lb BERT *small* uncased	1	6	12
Lb/De BERT *small* uncased	1	6	12
LuxemBERT *small* uncased	1	6	12
Lb BERT *medium* uncased	8	3	8
Lb/De BERT *medium* uncased	10	6	3
LuxemBERT *medium* uncased	11	5	3
Lb BERT *large* uncased	15	2	2
Lb/De BERT *large* uncased	17	2	0
LuxemBERT *large* uncased	**18**	**1**	**0**
mBERT cased	1	6	13
mBERT uncased	2	18	0

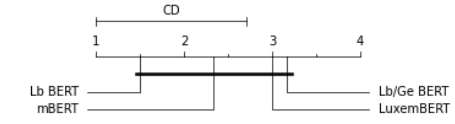
Table 7: Wins/ties/losses comparison, based on Wilcoxon superiority tests, of all models of this study.



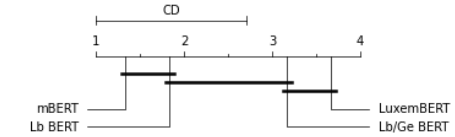
(a) *small* uncased



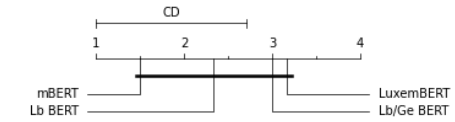
(b) *small* cased



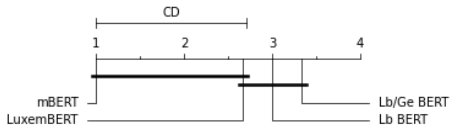
(c) *medium* uncased



(d) *medium* cased



(e) *large* uncased



(f) *large* cased

Figure 7: Comparison of mBERT, Lb BERT, Lb/De BERT, and LuxemBERT with Friedman/Nemenyi tests. An approach a is considered as significantly better than another (b) if its mean rank R_a is such that $R_a > R_b + CD$. The higher the mean rank, the better. These plots allow observing that the best approach is dependant on the size of the training data and the case. However, Lb/De and LuxemBERT are consistently among the best approaches. To decide which of the approach is the best in practice, we rely on Figure 7.

6. Threats to Validity

As all experimental, the work presented here can face potential threats to validity.

First, it is possible that the results of our experiments are contingent upon the *quantity* of data used in our experimental setup. To mitigate this risk, and to investigate the effect of data size on the approach we propose, we performed experiments with three different sizes of dataset.We also note that we leveraged new datasets to go beyond what was already available to the research community for the Luxembourgish language, thus enabling us to investigate three vastly different sizes of dataset.

The *quality* of the data could also threaten the strength of our conclusions. In particular, a lack of diversity in the training data would limit the performance of any language model. While some of the additional datasets we leveraged contain sentences of irregular quality (user comments), a significant part of our new datasets are made exclusively of high-quality, professionally written news articles.

When possible and meaningful, we computed statistical tests to measure the statistical significance of the performance difference of the tested approaches. Hence, it is possible to evaluate whether the observed differences are likely due to random fluctuations, or are more likely effects of the tested approaches.

7. Related Works

Over the last years, numerous BERT-like language models have been created. In particular, researchers trained and released models for wide-spread, Western languages such as German, French, and Spanish. Scheible et al. (2020) released GottBERT, a German model trained on the German portion of the OSCAR corpus (Suárez et al., 2020). A French BERT model was introduced by Martin et al. (2020) in the form of CamemBERT trained on the French portion of OSCAR. The Spanish version of BERT, BETO (Cañete et al., 2020) was trained on a combination of Spanish resources.

mBERT serves as an important language model for numerous less widespread languages as it offers versatility at the expense of performance. Indeed, Wu and Dredze (2020) compared mBERT's performance to that of monolingual baseline models on three NLP tasks. They showed that for low-resource languages such as Latvian or Mongolian, mBERT reached higher performances as opposed to monolingual models. The opposite was observed for models trained on high-resource languages.

Data augmentation for NLP has also been often studied: Kobayashi (2018) proposed to augment data by replacing words in given sentences by words with paradigmatic relations such as synonyms and antonyms for text classification tasks. Expanding on this approach, Wei and Zou (2019) leveraged synonym replacement, random insertion, random swap, and random deletion to further increase the performance of text classifiers. Liu et al. (2020) used data augmentation via conditional text generation based on a reinforcement learning model, which significantly boosted performances on three NLU tasks when compared to prior data augmentation techniques. Each of the aforementioned techniques augments data from the target language and creates synthetic sentences that are close to already existing data. On the contrary, our approach relies on authentic data from the auxiliary languages to enrich the dataset.

Wu and Dredze (2020) proposed a middle-ground

between mBERT and monolingual models for low-resource languages by training models on bilingual data. They suggested to pair them with a language that is closely related to the target language in order to increase the performance of the model. The resulting models outperformed the monolingual models on almost every selected task, however, they generally performed worse than mBERT. Our approach seeks to make the text data from the auxiliary language resemble the data written in the target language more closely.

8. Conclusion

In this paper, we introduced a new BERT model for Luxembourgish, a low-resource language. To circumvent the lack of data, we rely on two data augmentation strategies. We showed that they lead to improvement on six NLP tasks, even though it was not always possible to prove statistical significance between all variants. We showed that our Luxembourgish model, LuxemBERT, outperforms its only competitor, mBERT, in five of the six tested tasks. The cased LuxemBERT beats cased mBERT on all six tasks. In addition, we created Luxembourgish datasets for various NLP tasks, that we make available to researchers on request. We believe our work is a great addition to the NLP field, with a new BERT model for Luxembourgish and the release of four datasets. We also believe that our data augmentation strategy can be applied to other low-resource languages.

9. Acknowledgements

We would like to thank Dr. Christoph Purschke, Dr. Peter Gilles, and Daniela Gierschek (Institute for Luxembourgish Linguistics and Literature Studies / University of Luxembourg) as well as Tom Weber (Radio Télévision Luxembourg) and Dr. Caroline Döhmer (Zenter fir d'Lëtzebuerger Sprooch) for their continued support and for providing access to pre-training data, which made LuxemBERT possible.

10. Bibliographical References

Bernhard, D. and Ligozat, A.-L. (2013). Hassle-free pos-tagging for the alsatian dialects.

Carnie, A. (2021). *Syntax: A generative introduction*. John Wiley & Sons.

Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., and Pérez, J. (2020). Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. Academic press.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457.

Levesque, H., Davis, E., and Morgenstern, L. (2012). The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, R., Xu, G., Jia, C., Ma, W., Wang, L., and Vosoughi, S. (2020). Data boost: Text data augmentation through reinforcement learning guided conditional generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9031–9041.

Lothritz, C., Allix, K., Lebichot, B., Veiber, L., Bissyandé, T. F., and Klein, J. (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.

Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de la Clergerie, É. V., Seddah, D., and Sagot, B. (2020). Camembert: a tasty french language model. In *ACL 2020-58th Annual Meeting of the Association for Computational Linguistics*.

Ortiz Suárez, P. J., Romary, L., and Sagot, B. (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, July. Association for Computational Linguistics.

Pennebaker, J. W. (2011). The secret life of pronouns. *New Scientist*, 211(2828):42–45.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Scheible, R., Thomczyk, F., Tippmann, P., Jaravine, V., and Boeker, M. (2020). Gottbert: a pure german language model. *arXiv preprint arXiv:2012.02110*.

Suárez, P. J. O., Romary, L., and Sagot, B. (2020). A monolingual approach to contextualized word embeddings for mid-resource languages. In *ACL 2020-58th Annual Meeting of the Association for Computational Linguistics*.

Varrette, S., Bouvry, P., Cartiaux, H., and Georgatos, F. (2014). Management of an academic hpc cluster: The ul experience. In *Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014)*, pages 959–967, Bologna, Italy, July. IEEE.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (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.

Wei, J. and Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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)*, pages 6382–6388.

Wu, S. and Dredze, M. (2020). Are all languages created equal in multilingual bert? *ACL 2020*, page 120.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

11. Language Resource References

Goldhahn, Dirk and Eckart, Thomas and Quasthoff, Uwe and others. (2012). *Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages.*