PhD-FSTM-2023-032
The Faculty of Science, Technology and Medicine

# Dissertation

Defence held on 29/03/2023 in Luxembourg

to obtain the degree of

## Docteur de l'Université du Luxembourg en Informatique

by

Cedric LOTHRITZ
Born on 12th December 1990 in city of Luxembourg, (Luxembourg)

# NLP DE LUXE - CHALLENGES FOR NATURAL LANGUAGE PROCESSING IN LUXEMBOURG

## Dissertation Defence Committee

Dr Jacques KLEIN, dissertation supervisor
*Professor, University of Luxembourg*

Dr Tegawendé F. BISSYANDÉ, Chairman
*Professor, University of Luxembourg*

Dr Christoph PURSCHKE, Vice-Chairman
*Professor, University of Luxembourg*

Dr Jacques SAVOY, External Reviewer
*Professor, University of Neuchâtel*

Dr A. Seza DOĞRUÖZ, External Reviewer
*Professor, Ghent University*

# Abstract

The Grand Duchy of Luxembourg is a small country in Western Europe, which, despite its size, is an important global financial centre. Due to its highly multilingual population, and the fact that one of its national languages - Luxembourgish - is regarded as a low-resource language, this country lends itself naturally to a wide variety of interesting research opportunities in the domain of Natural Language Processing (NLP).

This thesis discusses and addresses challenges with regard to domain-specific and language-specific NLP, using the unique linguistic situation in Luxembourg as an elaborate case study. We focus on three main topics: (I) NLP challenges present in the financial domain, specifically handling personal names in sensitive documents, (II) NLP challenges related to multilingualism, and (III) NLP challenges for low-resource languages with Luxembourgish as the language of interest.

With regard to NLP challenges in the financial domain, we address the challenge of finding and anonymising names in documents. Firstly, an empirical study on the usefulness of Transformer-based deep learning models is presented on the task of Fine-Grained Named Entity Recognition. This empirical study was conducted for a wide array of domains, including the financial domain. We show that Transformer-based models, and in particular BERT models, yield the best performance for this task. We furthermore show that the performance is also strongly dependent on the domain itself, regardless of the choice of model. The automatic detection of names in text documents in turn facilitates the anonymisation of these documents. However, anonymisation can distort data and have a negative effect on models that are built on that data. We investigate the impact of anonymisation of personal names on the performance of deep learning models trained on a large number of NLP tasks.

Based on our experiments, we establish which anonymisation strategy should be used to guarantee accurate NLP models.

Regarding NLP challenges related to multilingualism, we address the need for polyglot conversational AI in a multilingual environment such as Luxembourg. The trade-off between a single multilingual chatbot and multiple monolingual chatbots trained on Intent Classification and Slot Filling for the banking domain is evaluated in an empirical study. Furthermore, we publish a quadrilingual, parallel dataset that we built specifically for this study, and which can be used to train a client support assistant for the banking domain.

With regard to NLP challenges for the Luxembourgish language, we predominantly address the lack of a suitable language model and datasets for NLP tasks in Luxemborgish. First, we present the most impactful contribution of this PhD thesis, which is the first BERT model for the Luxembourgish language which we name LuxemBERT. We explore a novel data augmentation technique based on partially and systematically translating texts to Luxembourgish from a closely related language in order to artificially increase the training data to build our LuxemBERT model. Furthermore, we create datasets for a variety of downstream NLP tasks in Luxembourgish to evaluate the performance of LuxemBERT. We use these datasets to show that LuxemBERT outperforms mBERT, the de facto state-of-the-art model for Luxembourgish. Finally, we compare different approaches to pre-train BERT models for Luxembourgish. Specifically, we investigate whether it is preferable to pre-train a BERT model from scratch or continue pre-training an already existing pre-trained model on new data. To this end, we further pre-train the multilingual mBERT model and the German GottBERT on the Luxembourgish dataset that we used to pre-train LuxemBERT and compare all models in terms of performance and robustness. We make all our language models as well as the datasets available to the NLP community.

*Take every chance,*
*Take every second chance seriously.*

# Acknowledgements

This four-year-long journey to complete this thesis would never have been possible if it were not for the support and help of many people to whom I am deeply grateful.

First, I would like to thank my supervisor, Prof. Jacques Klein, for giving me the opportunity to pursue an academic career. I am deeply grateful for his continued advice, help, and support which allowed me to realise my potential as a researcher and hone my skills. I am also thankful for his trust in me, even in times when I doubted myself.

I am just as thankful to my co-supervisor, Prof. Tegawendé Bissyandé, for continually advising me, helping me develop my scientific thinking and paper writing skills, and for teaching me how to assess my own research critically.

Third, I would like to express my thanks to Prof. Christoph Purschke, for being on the defense committee as Vice-Chair, and his help in my research projects related to Luxembourgish NLP which made that particular part of my thesis a great success.

I would also like to express my gratitude to the remaining members of my defense committee, Prof. Jacques Savoy, Prof. A. Seza Doğruöz, and Dr. Andrey Boytsov. It is a great honour to have them on my defense committee, and I am grateful to them for taking the time to read and assess this dissertation.

Then, I would like to thank my many co-authors and advisors at the University of Luxembourg, including Prof. Kévin Allix, Dr. Bertrand Lebichot, Ms. Lisa Veiber, and Dr. Saad Ezzini, for all of their great help in making my PhD a success.

I am as well grateful to my scientific advisors at BGL BNP Paribas, including Mrs. Anne Goujon, and Mr. Clément Lefebvre for their ideas and help in my research and giving me the opportunity to bridge the gap between the academic world and the industry.

I also thank my two former interns, Ms. Nabila Khouya and Ms. Isabella Olariu, for giving me the opportunity to act as an advisor myself, allowing me to pass my knowledge onto them and help them successfully complete their Master's theses.

I want to express my thanks to my colleagues and the friends I made in the TruX and SerVal research groups at SnT, for all the discussions and memories we made during that time.

I would like to thank my family and friends who supported my choice to pursue a PhD degree, and have always and believed in me during this journey, above all my Mum, who was always there for me, especially during the most difficult of times.

Finally, I want to thank my dog for his emotional support and for making sure that I get regular exercise.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

*In this first chapter, we introduce the general motivation for researching domain-specific and language-specific NLP problems present in Luxembourg. We then present specific challenges in three parts, followed by our contributions addressing those challenges. Finally, we show the roadmap for the remainder of this thesis.*

## Contents

## 1.1 Motivation

The Grand Duchy of Luxembourg is a small country in Central Europe, surrounded by France, Germany, and Belgium. Despite its size, the country attracts myriad international companies and investors, resulting in a strong economy, not only by European standards, but by global standards as well.

Organisations settling in Luxembourg are typically faced with the task of adapting themselves to the unique linguistic situation in the country and the varied demographic makeup of its population. Both of these factors give rise to numerous challenges with regard to automation of tasks, in particular of those involving the automated processing of documents and textual data in general. This in turn makes Luxembourg a magnet for interesting research questions in the domain of Natural Language Processing (NLP).

One of the most important branches of the economy in Luxembourg is the financial sector. Indeed, it makes up nearly a third of the country's GDP and the Luxembourg Banker's Association (ABBL) represents 123 banks in Luxembourg[1], with the Deutsche Bank, the Banque et Caisse d'Épargne de l'État, and BGL BNP Paribas being some of the most important banks in the country. Table 1.1 shows the ten most important banks in Luxembourg in terms of total assets[2].

Table 1.1: Ten largest banks in Luxembourg in terms of total assets (in million Euros)

| Name | Business Focus | Assets |
|---|---|---|
| JP Morgan Bank Luxembourg SA | Investment Banking | 66 880 |
| Banque et Caisse d'Épargne de l'État | Universal Banking | 53 764 |
| BGL BNP Paribas | Universal Banking | 51 642 |
| Société Générale Luxembourg | Private B., Investment B. | 51 333 |
| CACEIS Bank, Luxembourg Branch | Securities Services | 46 082 |
| Banque Internationale à Luxembourg | Universal Banking | 32 445 |
| Deutsche Bank Luxembourg SA | Wealth Management | 27 530 |
| ING Luxembourg | Universal Banking | 23 303 |
| Intesa Sanpaolo Bank Luxembourg SA | Corporate Banking | 21 091 |
| DZ Privatbank SA | Private Banking | 20 915 |

With the continuous growth of financial institutes, the accompanying challenges are also increasing in number. These challenges include the rising number of financial documents to be treated, the ever-present risk of defrauding schemes, and the increasing concerns of data privacy. Evidently, there is a vested interest in investigating automated approaches to handle the growing workload, as it becomes progressively infeasible to be managed by humans. As such, the deployment of NLP models that can handle tedious, yet simple, tasks with little to no oversight can significantly reduce the need for manual labour. However, financial documents, by their nature, present a number of challenges that make the creation of adequate NLP models more difficult. For one, they are examples of non-typical textual data with specific use of

[1] `https://www.abbl.lu/en/home`

[2] according to: `https://thebanks.eu/banks-by-country/Luxembourg`

language and specialised vocabulary. These domain-specific linguistic characteristics could reduce the performance of off-the-shelf models.

Another factor that makes the deployment of NLP models in Luxembourgish organisations more challenging, is the diversity of the country's population. Luxembourg is marked by a high number of border workers, as well as high immigration, making its diversity and multiculturalism immediately apparent. The demographic situation in Luxembourg stems from a population shift with a continuous increase of the share of foreigners since the 1960s and 1970s [6][7]. Nowadays, the population of Luxembourg is made up of more than 175 different nationalities despite its small population size of almost 650 000, with most people being either of Luxembourgish ($\simeq$ 52.88%), Portuguese ($\simeq$ 15.55%), French ($\simeq$ 7.6%), or Italian ($\simeq$ 3.74%) descent. Table 1.2 shows the demographic makeup of Luxembourg[3]. With multiculturalism comes multilingualism which is evident by the multitude of languages spoken in the country. Influenced by its neighbouring countries, Luxembourg's administrative languages are French, German, and Luxembourgish, the latter being a Moselle-Franconian dialect closely related to German [8]. With the addition of English, Luxembourgers learn each of these languages in school and use them largely on a daily basis. With such a diverse pool of languages spoken by potential clients and business partners, companies are incentivised to invest into managing multilingual systems to accommodate a userbase comprised of people who do not all speak the same language at the same degree of proficiency.

Table 1.2: Demographic makeup of the population in Luxembourg as of January 1, 2022

| Demographic | Size | % of population |
|---|---|---|
| Luxembourgish | 356 775 | 52.88 |
| Portuguese | 93 678 | 15.55 |
| French | 49 173 | 7.60 |
| Italian | 24 116 | 3.74 |
| Belgian | 19 414 | 3.01 |
| German | 12 796 | 1.98 |
| Spanish | 8388 | 1.30 |
| Romanian | 6405 | 0.99 |
| Polish | 5020 | 0.78 |
| Chinese | 4142 | 0.64 |
| British | 4104 | 0.64 |
| Dutch | 4069 | 0.63 |
| Greek | 4017 | 0.62 |
| Asian | 19 066 | 2.95 |
| African | 13 668 | 2.12 |
| American | 7707 | 1.10 |
| Oceanian | 244 | 0.04 |

A final factor that hinders the creation of NLP models in Luxembourg relates to the Luxembourgish language itself. Originating as a Moselle Franconian dialect before

[3] https://www.justarrived.lu/en/practical-information/population-in-luxembourg/

being formally recognised as an official language [6], Luxembourgish is spoken by nearly 600 000 people world-wide[4]. Due to the small number of native speakers, the prevalence of other languages, and the fact that it is predominantly a spoken language, textual data written in Luxembourgish is sparse compared to text written in widespread languages such as English, Spanish, or Chinese. This is further illustrated by the fact that printed media such as newspapers and magazines are typically not available in Luxembourgish, but in German or French, adding to the scarcity of textual data in Luxembourgish. Additionally, the number of articles on the Luxembourgish Wikipedia, which is 61 159 at the time of writing, exceeds the median number of Wikipedia articles per language, which is 9 497. However, this amounts to fewer than half a million sentences, which is comparably low. Due to this lack of data, Luxembourgish is considered a low-resource language, posing a challenge for working on NLP problems in Luxembourgish.

In this work, we will discuss various domain-specific and language-specific challenges present in the NLP field. Due to the importance of the financial domain and the aforementioned unique linguistic circumstances, the country of Luxembourg acts as a suitable case study for this thesis. We will focus on three major aspects important to Luxembourg: (I) NLP in the Financial Domain, (II) Multilingualism, and (III) Luxembourgish NLP. We aim to determine and address specific challenges relevant to each of these aspects, in order to strengthen the Luxembourgish NLP community. In a partnership with the Luxembourgish bank BGL BNP Paribas, we determined a variety of research problems based on actual use cases and projects. In return, our research helped our partners make decisions to improve their workflows, revealing a tangible benefit of the NLP community on the industry and bridging the gap between academia and the industry.

## 1.2 Challenges

In this section, we present relevant challenges for each domain we address in this work: challenges in the financial domain, challenges stemming from multilingualism, and challenges regarding the low-resource nature of the Luxembourgish language.

### 1.2.1 Challenges Related to the Financial Domain

The main challenges for financial institutions lie in the sensitive nature of the data being processed. Mistakes and failure to comply with regulations can have catastrophic financial and legal consequences. Furthermore, textual data must not leave the organisation, so they cannot rely on external services and are thus limited to their own resources.

#### 1.2.1.1 Detection of Names

Documents processed at financial institutions, such as letters of credit or loan applications, need to be analysed with regard to the names they contain, e.g., the names of loan applicants, company names involved in trade deals, brand names, etc. This kind of information is crucial, e.g., to anonymise sensitive information, to determine if given trade partners comply with law and trade regulations, and to ensure that information is consistent across documents. While models trained on Named Entity Recognition exist, it is not guaranteed that the existing models

[4] according to `https://cursus.edu/en/23040/luxembourgish-at-its-best`

work on these kinds of data, as there is a significant difference with regard to the language used in financial documents. Not only is there a difference from a vocabulary standpoint, but also in terms of style, requiring models that are specialised in the financial domain.

#### 1.2.1.2 Protection of Personal Data

Financial institutes have to handle vast amounts of data that contain sensitive information such as clients' names, postal addresses, phone numbers, etc. It is crucial to keep this information protected from leaving the system. This is true not only for banks but for any organisation that stores personally identifiable information from their customers. Since the introduction of the General Data Protection Regulation (GDPR)[9], companies have been under increased scrutiny regarding the collection, storage, and processing of personal data. Under the threat of sizeable fines, companies are incentivised to keep such data protected, and anonymised before further processing it. While it is relatively easy to find and anonymise such information in structured data formats such as tables, unstructured texts such as legal documents, loan applications, and emails pose a challenge for this task. Not only is the detection of proper names more difficult in natural text, but the act of anonymisation can have a detrimental effect on the usability of the text in downstream NLP tasks.

### 1.2.2 Challenges Related to Multilingualism

Multilingual systems are evidently more demanding than monolingual ones, and the effort required to manage a multilingual system is proportional to the number of languages needed. For example, company websites hosted in a country such as the UK are typically displayed in a single language, whereas a similar website in a country such as Switzerland, Luxembourg, South Africa, or India, should ideally cover every official language in order to reach most of the country's population. However, while websites that are simply tasked to display information are relatively easy to expand to include several languages, complex tasks that require one to interact with the userbase are not. Such tasks include holding conversations, analysing reviews, or automatically dispatching incoming emails. These simple tasks are usually handled by automated systems powered by machine learning or deep learning models. The multilingual setting introduces a new set of obstacles. For instance, there might not be sufficient training data for each language, leading to some languages being under-represented and, in turn, leading to a worse performance of the resulting system. In addition, multilingual areas tend to lead to regular code switching. People who speak multiple languages are likely to switch between languages while speaking or writing, increasing the challenge for multilingual systems that actively interact with their users [10][11]. This can also be frequently observed in Luxembourg. While the country has three official languages (Luxembourgish, German, and French), there are no well-defined language regions. Unlike Belgium and Switzerland where a given language is predominantly used depending on the geographical area, the lines are blurred in Luxembourg [6]. Typically, native Luxembourgers speak Luxembourgish among themselves regardless of the region and switch to German, French, or English when speaking to non-natives or foreigners.

### 1.2.3 Challenges Related to Luxembourgish NLP

As a low-resource language, the challenges related to Luxembourgish typically involve the lack of high-quality textual data. However, there are additional factors that exacerbate the issues with texts written in Luxembourgish, including data scarcity and quality.

#### 1.2.3.1 Scarcity of Annotated Data

In order for ML models to adequately solve specific NLP problems, they need to be trained on large datasets of labelled data. As there is a high focus of training models for the English language, there is no shortage of datasets at hand. However, this is rarely the case for low-resource languages. Datasets should ideally be annotated manually and by multiple native speakers to increase inter-annotator agreement and ensure high-quality data. While it is not difficult to find many native speakers of wide-spread languages, it is a considerable challenge for languages spoken by few people.

#### 1.2.3.2 Scarcity of Unannotated Data

For many languages, the scarcity of data does not only mean a lack of annotated data for supervised learning tasks, but a shortage of textual data in general. Modern language models such as BERT models [3] require a vast amount of data which can be difficult to find.

#### 1.2.3.3 Data Quality

The problem of scarcity in case of the Luxembourgish language is exacerbated by several factors that can lead to spelling variations of words and noisy textual data. Table 1.3 shows various valid spellings of personal pronouns in Luxembourgish. Most of these words have two valid spellings which both need to be learned by a language model. These variations can have various reasons. We differentiate between three types of variations: regional variations, grammatical/sociolinguistic variations, and historical variations [8]:

**Regional Variations**
There are several Luxembourgish dialects despite the small size of the country [12]. As such, most words have multiple valid spellings that differ depending on the speaker's/writer's region of origin. These differences in spelling can be small with a single letter that differs between spelling variations; however, oftentimes, entirely different words are used depending on the dialect. For instance, Table 1.3 shows that there are two possible variations for the word "us": "eis" and "ons".

**Grammatical/Sociolinguistic Variations**
Despite Luxembourgish being an officially recognised language with standardised grammar and spelling rules, it is primary a spoken language rather than a written one, although this is slowly changing [13][7]. Furthermore, compared to German, French, and English, the language is barely taught at schools in Luxembourg [8]. Both of these factors lead to most people struggling to write Luxembourgish in accordance with official grammatical rules despite being native speakers [14]. This in turn leads to a large number of spelling variations for most words in text sources such as text messages, forums, social media, or blogs [13] which can make them undesirable for building generative language models that are supposed to write grammatically

correct text.

### Historical Variations

Finally, Luxembourgish has been subject to multiple spelling reforms, which typically lead to significant orthographic changes in many words [7][14]. The last spelling reform came into effect in 2019, leading to numerous problems for NLP models for the Luxembourgish language. The biggest such problem concerns texts that were written before the reform which now contain a lot of invalid spelling variations. This, similarly to texts with a lot of grammatical variations, severely limits the amount of text that can be used for grammatically correct generative language models.

Table 1.3: Spelling variations of personal pronouns in Luxembourgish[5]

| Number | Person | Gender | Nominative | Accusative | Dative |
|---|---|---|---|---|---|
| Singular | 1st | - | ech | mech | mir/mer |
| | 2nd | - | du/de | dech | dir/der |
| | 3rd | Male<br>Neutral<br>Female | hien/en<br>hatt/et/’t<br>si/se | hien/en<br>hatt/et/’t<br>si/se | him/em<br>him/em<br>hier/er |
| Plural | 1st | - | mir/mer | eis/ons | eis/ons |
| | 2nd | - | dir/der | iech | iech |
| | 3rd | - | si/se | si/se | hinnen/en |

#### 1.2.3.4 Multilingual Texts

In addition to the inconsistent spelling of words, Luxembourgish texts also oftentimes contain inconsistent use of languages, i.e. textual data tends to be a mixture of multiple languages, specifically in news articles and press releases. Since Luxembourgish people are generally multilingual, French and German text is typically not translated in news media. This can include interviews with non-Luxembourgish people, or official communiques which are oftentimes written in French.

## 1.3 Contributions

The contributions of this work to address the aforementioned challenges can be summarised as follows in three major parts:

In the first part, we address the challenges present in the financial domain. In particular, we focus on the handling of names in financial and other sensitive documents. In a first step, we tackle the challenge of recognising names in a document and studying the effect of the domain on the performance of the examined models. In a second step, we study the effect of anonymising personal names on the subsequent processing of a document.

- *A comparison of Transformer based models on fine-grained Named Entity Recognition*: Named Entity Recognition (NER) is a fundamental Natural Language Processing (NLP) task and has remained an active research field. In recent years, Transformer models and more specifically the BERT model developed at Google revolutionised the field of NLP. While the performance of Transformer-based approaches such as BERT has been studied for NER, there has not yet been a study for the fine-grained Named Entity Recognition

(FG-NER) task. In this chapter, we compare three Transformer-based models (BERT, RoBERTa, and XLNet) to two non-Transformer-based models (CRF and BiLSTM-CNN-CRF). Furthermore, we apply each model to a multitude of distinct domains. We find that Transformer-based models incrementally outperform the studied non-Transformer-based models in most domains with respect to the F1 score. Furthermore, we find that the choice of domain significantly influenced the performance regardless of the respective data size or the chosen model.

- *A comparison of various anonymisation strategies on the impact of downstream NLP tasks*: Data anonymisation is often required to comply with regulations when transferring information across departments or entities. However, the risk is that this procedure can distort the data and jeopardise the models built on it. Intuitively, the process of training an NLP model on anonymised data may lower the performance of the resulting model when compared to a model trained on non-anonymised data. In this chapter, we investigate the impact of anonymisation on the performance of nine downstream NLP tasks. We focus on the anonymisation and pseudonymisation of personal names and compare six different anonymisation strategies for two state-of-the-art pre-trained models. Based on these experiments, we formulate recommendations on how the anonymisation should be performed to guarantee accurate NLP models.

In the second part, we study the challenges in multilingual systems. Specifically, we evaluate to what degree the presence of multiple languages affects the performance of such systems.

- *An empirical study to compare monolingual and multilingual chatbots:* With the momentum of conversational AI for enhancing client-to-business interactions, chatbots are sought in various domains, including FinTech where they can automatically handle requests for opening/closing bank accounts or issuing/terminating credit cards. Since they are expected to replace emails and phone calls, chatbots must be capable to deal with diversities of client populations. In this chapter, we focus on the variety of languages, in particular in multilingual countries. Specifically, we investigate the strategies for training deep learning models of chatbots with multilingual data. We perform experiments for the specific tasks of Intent Classification and Slot Filling in financial domain chatbots and assess the performance of mBERT multilingual model vs multiple monolingual models.

In the third part, we address the challenges related to the low-resource nature of the Luxembourgish language. In a first step, we focus on mitigating the lack of textual data by examining the usefulness of a novel data augmentation scheme for the creation of a Luxembourgish language model. In a second step, we examine the trade-offs of various pre-training schemes and use the gained knowledge to improve upon our Luxembourgish language model. We also mitigate the lack of annotated data by providing numerous Luxembourgish datasets for NLP tasks to the community.

- *A first Transformer-based language model for the Luxembourgish language*: In this chapter, 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 as 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 them available to researchers on request.

- *A comparison of various pre-training schemes for Luxembourgish language models*: In this chapter, 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.

## 1.4 Roadmap

Figure 1.1 shows the overall structure of this work. The general introduction is followed by Chapter 2, which provides a general background to facilitate understanding the rest of the thesis. The main part of the thesis will focus on the three pillars that mark interesting research directions for NLP in Luxembourg: (I) Financial NLP in Chapters 3 and 4, (II) Multilingualism in Chapter 5, and (III) Luxembourgish NLP in Chapters 6 and 7. Finally, we will conclude this dissertation and address future work in Chapter 8.

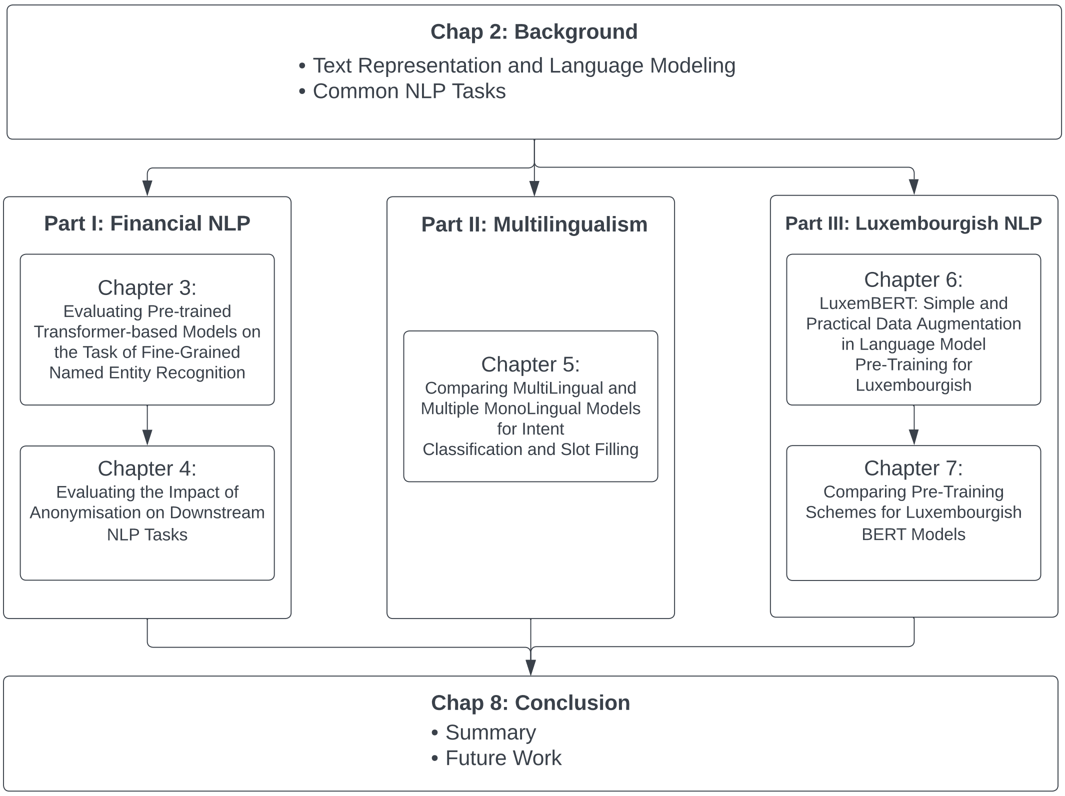

Chap 2: Background
• Text Representation and Language Modeling
• Common NLP Tasks
Part I: Financial NLP
Chapter 3: Evaluating Pre-trained Transformer-based Models on the Task of Fine-Grained Named Entity Recognition
Chapter 4: Evaluating the Impact of Anonymisation on Downstream NLP Tasks
Part II: Multilingualism
Chapter 5: Comparing MultiLingual and Multiple MonoLingual Models for Intent Classification and Slot Filling
Part III: Luxembourgish NLP
Chapter 6: LuxemBERT: Simple and Practical Data Augmentation in Language Model Pre-Training for Luxembourgish
Chapter 7: Comparing Pre-Training Schemes for Luxembourgish BERT Models
Chap 8: Conclusion
• Summary
• Future Work

Figure 1.1: Roadmap for this thesis

# 2 Background

*In this chapter, we present previous work that forms the basis for this thesis. We will present the evolution of text representations for NLP models in general, but we will also highlight work that was done for multilingual and low-resource settings, which is crucial for NLP in a country such as Luxembourg. Furthermore, we will present NLP tasks that will be commonly featured throughout this work.*

## Contents

## 2.1 Text Representation and Language Modeling

One crucial aspect to solve any Machine Learning task to a satisfying degree is to create an appropriate representation of the input data fed to the model. While this aspect can be straightforward depending on the nature of the data, the representation of textual data for NLP models is, in and of itself, a challenge for solving NLP problems adequately. Words carry meaning that humans can understand with little mental effort; machines, on the other hand, do not process text in the same way. As such, a first step to handle tasks in this field is to convert textual data into a form that a machine can understand. Throughout the past decades, there have been numerous ways to represent text in the form of numerical vectors. Those representations can largely be divided into three categories: Frequency-based vectors, static word embeddings, and embeddings derived from language models.

### 2.1.1 Frequency-Based Approaches to Text Representation

Many classical approaches to text representation revolve around the frequency of words present in the text. Being straightforward and light on resources, they can be useful for classification and topic modeling tasks. The most well-known techniques include **B**ag-**O**f-**W**ords (BOW), bag of n-grams [15], and **T**erm **F**requency-**I**nverse **D**ocument **F**requency (TF-IDF) [16] [17]:

Given a vocabulary $V$ of size $|V| = k$ and a word sequence $s$:

BOW vectors are $k$-dimensional vectors of the form

$$v(s) = \langle f_1, f_2, ..., f_k \rangle$$

where $f_i$ denotes the frequency of the $i$th word in $V$ in word sequence $s$. This kind of representation is useful to encode the general idea of a text sequence, making it an effective technique for simple text categorisation and topic modeling problems.

In contrast to the BOW model, the n-gram model encodes word sequences as subsequences of size $n$, allowing to capture the word order of the sequence. To a certain degree, it also allows to more effectively encode ambiguous words that have different meanings depending on the group of words they are part of such as:

...the United States of America...
... water has three states of matter...

Here, an NLP model could use trigrams to differentiate between "States of America" and "states of matter", which would not be possible with a BOW approach.

TF-IDF is a weighting approach that improves on BOW models by encoding the importance of a specific word (here: *term*) for a given word sequence (here: *document*) in a corpus. For instance, common terms such as the determiner "the" are generally not very relevant in a given document even if they are very frequent. TF-IDF weighting reduces the importance of such terms by dividing their frequency in the document by their frequency in the entire corpus. A TF-IDF vector for a given document $d$ with vocabulary $V$ and corpus $C$ is given by:

$$v(d) = \langle tf(t_1, d) \times idf(t_1, C),...,tf(t_{|V|}, d) \times idf(t_{|V|}, C) \rangle$$

$$\text{with} \\ tf(t_i, d) = \frac{f_{t_i,d}}{\sum_{t_j \in V} f_{t_j,d}} \\ idf(t_i, C) = log\frac{|C|}{|\{d_j \in C : t_i \in d_j\}|}$$

where $f_{ti,d}$ denotes the frequency of the $i$th term in $V$ in document $d$. This allows models to effectively ignore common words such as stop words and function words that could otherwise impact the model's performance.

While these techniques are easy to implement, and do not present a strain on resources, they do not take into account the underlying meaning of the words. In addition, BOW and TF-IDF also ignore the word order and context in a given sequence. For example, the sentences "You owe the banks a lot of money." and "The banks owe you a lot of money." would have identical BOW and TF-IDF representations despite having opposite meanings. While n-grams do represent text as ordered chunks of words, they only mitigate this issue to a small degree, taking into account only the direct context of any given word. Finally, neither approach can encode *Out of Vocabulary* (OOV) words, limiting the number of words that can be represented in a vector.

In summary, these approaches are severely limited in their usefulness due to their lack of properly encoding semantic information of the text or word context. Thus, for many NLP tasks such as Natural Language Generation (NLG) and Natural Language Understanding (NLU) tasks, these techniques are inadequate.

### 2.1.2 Static Word Embeddings

In an attempt to encode the meaning of words in a fixed-size numerical vector, Mikolov et al. proposed two architectures to learn embeddings of words from their contexts using a huge text corpus: Continuous bag-of-words (CBOW) and skip-gram [1] (cf. Figure 2.1). In the CBOW model, a word $w_t$ is predicted from its context words $\langle w_{t-k}, .., w_{t-1}, w_{t+1}, ..., w_{t+k}\rangle$ with context window $k$. In contrast, the Skip-gram predicts the context around a given word $w_t$. Words which appear in similar contexts are ideally mapped to a similar representation in the vector space.

Such word embeddings can then be used as features for ML models, significantly outperforming simple frequency-based vectors. The most popular word embeddings based on the architectures introduced by Mikolov et al. include word2vec [18], GloVe [19], and fastText [20].

While these approaches are a vast improvement over the ones presented in Section 2.1.1, they are not without drawbacks. Their main disadvantage is that for any given word, there is only a single vector representation. As such, words with several possible meanings such as "hand" (which could refer to a human hand or the hand on a clock) will be encoded to the same vector regardless of the context they appear in. In addition, word embeddings do not solve the word order problem present in frequency-based vectors as sentence vectors are typically constructed by averaging word embeddings without considering the order of the words in the sentence. Finally, the OOV problem that is prevalent in frequency-based approaches, is also present for both the word2vec and GloVe techniques. This issue is addressed in the fastText approach which trains representations for subwords rather than entire words.

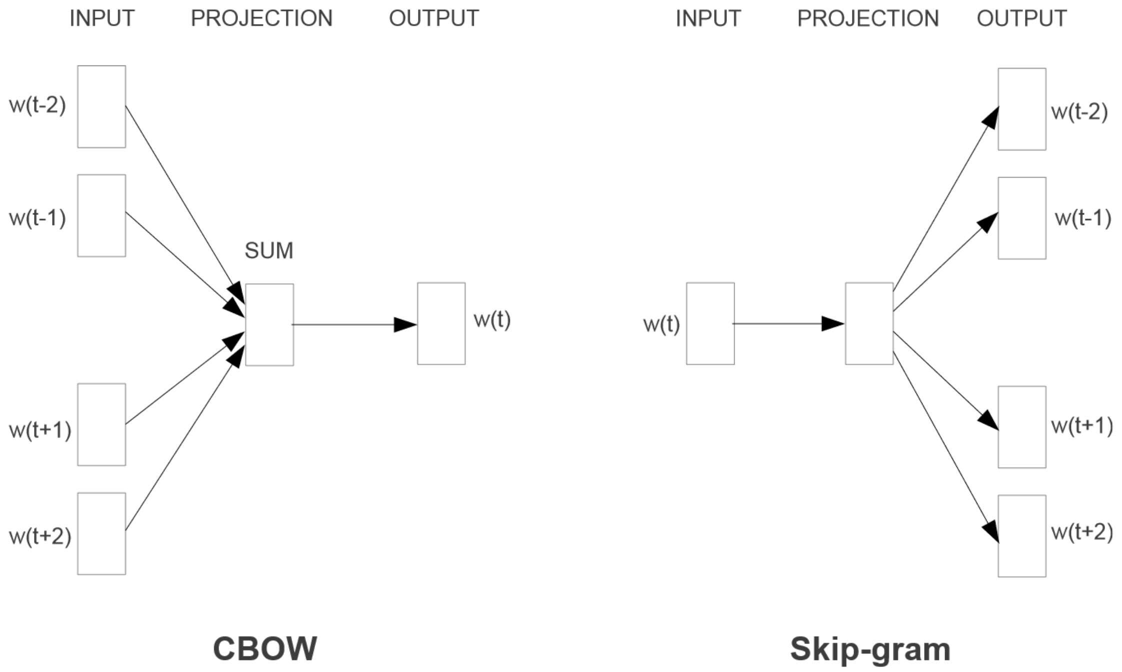


Figure 2.1: CBOW and Skip-gram models proposed by Mikolov et al. [1]

### 2.1.3 Word Embeddings from Language Models

Language Models (LM) are probabilistic models trained to assign probabilities to word sequences and predict missing words in a sequence [21]. This allows for creating different embeddings for words depending on their context, e.g. consider the following sentences:

I hurt my hand in an accident.
The clock's small hand points to 12.

The meaning of the word "hand" can be determined by the context words, allowing for a model to distinguish between the different possible meanings of ambiguous words. As such, LMs address the word order and ambiguity problems faced by classic approaches and static word embedding techniques such as word2vec. Outputs of LMs can either be used as embeddings [22] for ML models, or be fine-tuned for specific supervised problems [23].

One early architecture for context-sensitive LMs is **E**mbeddings from **L**anguage **Mo**dels (ELMo) by Peters et al. [22], which significantly advanced the state of the art in the NLP field. By combining a left-to-right and a right-to-left LSTM [24] network, words can be modeled as a concatenation of functions of each network's internal states. This allows to create vector representations that take into account context words from both directions. The authors showed that neural networks that used ELMo embeddings outperformed state-of-the-art models for six important NLP tasks including SQuAD for Question Answering [25] and SST-5 for Sentiment Analysis [26].

#### 2.1.3.1 The Rise of Transformers

Vaswani et al. [2] first described the Transformer model which superseded the popular LSTM model in favour of the attention mechanism [27] . Figure 2.2 shows the Transformer architecture in detail. The Transformer consists of stacks of $N$ identical encoder blocks and decoder blocks. Each encoder block consists of a multi-head self-attention layer connected to a fully connected feed-forward neural network. This attention layer helps the model put focus on other words in the input sequence while encoding a given word. Along with these components, decoder blocks feature an additional attention layer focusing on the previous output sequence of the model.

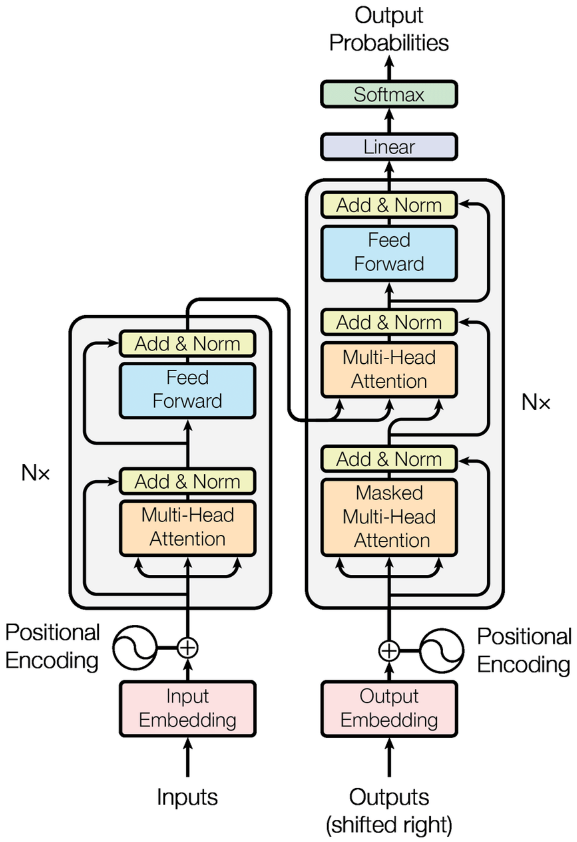


Figure 2.2: The architecture of the Transformer model [2]

Unlike Recurrent Neural Networks (RNN) such as LSTMs, Transformers do not need to process words in sequence, instead, entire sentences can be processed at once. This allows for more parallelisation than RNN models, and in turn, Transformer models can be trained much faster. Due to this advantage, Transformers have become fundamental for state-of-the-art models in the NLP field.

One early notable model that employed Transformers is the Generative Pre-training Transformer (GPT) model [28]. It was trained on a Next Token Prediction task, using a huge corpus. The original GPT model outperformed state-of-the-art models in nine out of twelve NLU tasks. However, arguably the most well-known and popular Transformer-based models are BERT and BERT-like models. [3]

#### 2.1.3.2 The Introduction of BERT

Devlin et al. revolutionised the NLP landscape by introducing **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (BERT) [3]. Three factors helped BERT become a state-of-the-art model and shape the future of the NLP domain for years to come: its architecture, its pre-training algorithms, and its pre-training corpus size.

Unlike the unidirectional GPT model or the pseudo-bidirectional ELMo model, BERT is a jointly bidirectional Transformer model, which allows it to capture context from the left-to-right and right-to-left direction. Figure 2.3 shows the difference between these three architectures. This in turn helps encode the meaning of ambiguous words using the context words around it.

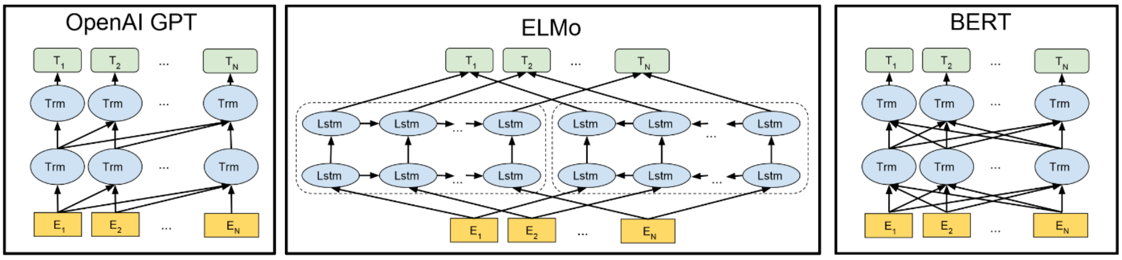


Figure 2.3: Architecture differences between ELMo, GPT, and BERT (adapted from Devlin et al [3])

BERT is pre-trained on two learning algorithms: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The objective of MLM is to learn the relationship between words. Specifically, MLM is a 2-step process where the first step consists of randomly masking out words from a given sentence. In a second step, the model attempts to reconstruct the original sentence by predicting the words that were masked out:

After Step 1: I [MASK] 200 euros at the [MASK].
After Step 2: I withdrew 200 euros at the bank.

On the other hand, NSP is a sentence pair classification task, allowing the model to learn the relationship between sentences. Specifically, given a sentence pair A and B from a corpus, the task consists of predicting whether B is preceded by A in the corpus:

A: I went to the bank. B: I withdrew 200 euros. *is_next*
A: I went to the bank. B: The dog wagged its tail. *is_not_next*

Finally, the model was trained on a large corpus comprising the complete English Wikipedia and the BooksCorpus [29], adding up to 130 million sentences or 13GB of textual data.

Devlin et al. pre-trained two different BERT models: *BERT Base* and *BERT Large.* In order to compare its performance to the original GPT model, *BERT Base* contains 12 Transformer blocks, 768 hidden layers, 12 self-attention blocks, and 110 million learnable parameters. The *Large* model is comprised of 24 Transformer blocks, 1024 hidden layers, and 16 self-attention blocks, for a total of 340 million trainable parameters.

Fine-tuned *BERT Base* and *BERT Large* models outperformed the state of the art in eleven tasks, including the GLUE [30] and SQuAD [25] benchmarks.

#### 2.1.3.3 The Ubiquity of BERT-like Models

Recent years have marked the appearance of a wide array of models that expanded or improved on the original BERT model, leading to numerous models outperforming the original model on many NLP tasks including the GLUE benchmark. Indeed, at the time of writing (December 2022), the BERT Large model is at position 47 on the GLUE leaderboard[1].

Some notable examples of such models include XLNet [31] (Chapter 3), RoBERTa [32] (Chapter 3), and ERNIE [33] (Chapter 4).

Yang et al. introduced XLNet [31], replacing the MLM task of the BERT model with a permutation-based autoregression task, effectively predicting sentence tokens in random order. Furthermore, they vastly increase the pre-training corpus size from 13GB to 158GB. XLNet manages to outperform BERT Large in 20 tasks, including the GLUE, SQuAD and RACE [34] benchmarks.

Liu et al. introduced **Ro**bustly optimized **BERT a**pproach (RoBERTa). [32] Similarly to XLNet, they used a larger pre-training corpus, consisting of 160GB of textual data, and they trained the model for longer periods of time. In addition, the authors tweaked the MLM pre-training task by dynamically applying masks to the sentences for every iteration, and removed the NSP task. The authors reported that RoBERTa outperforms both XLNet and BERT Large on the GLUE, SQuAD, and RACE benchmarks.

Finally, Sun et al. introduced **E**nhanced **R**epresentation through k**N**owledge **I**nt**E**gration (ERNIE) [33]. They further tweak the MLM task by adding phrase-level masking and entity-level masking, masking out entire groups of words and named entities from the pre-training sentences, respectively. Furthermore, they add numerous word-level and sentence-level tasks to the pre-training phase of their model. They report that the ERNIE model outperforms both BERT and XLNet on 16 tasks, including the GLUE benchmark.

#### 2.1.3.4 Domain-Specific and Non-English Language Models

While most modern language models generally yield a high performance for many NLP tasks, they typically have major drawbacks. First, they perform less well on domain-specific tasks [35]. Many domains such as the legal, the financial, or the scientific domain have specialised vocabularies with words that either barely or not at all appear in most corpora. As such, there has been considerable effort to train models on dedicated datasets. Examples of such models include FinBERT trained on financial texts [36], LEGAL-BERT trained on legislative text and court case documents [37], and SciBERT trained on scientific papers from various fields of study.

Another drawback of most language models such as BERT is that they are trained on

[1] `https://gluebenchmark.com/leaderboard`

data written entirely in English, making them less usable for non-English languages. In recent years, BERT models for every wide-spread language have been published. Large text collections that cover multiple languages such as Wikipedia or the OSCAR corpus [38][39] proved to be useful for delivering the necessary textual data for models such as the French CamemBERT [40], the German GottBERT [41], or the Spanish BETO [42].

In order to mitigate the restrictions of monolingual LMs, various multilingual models have been introduced, most notably multilingual BERT (mBERT) (Chapters 5, 6, and 7) which was jointly pre-trained on the 100 largest Wikipedia corpora, covering numerous low-resource languages including Luxembourgish. While the model is indeed useful for performing NLP tasks in a language for which no LM is available, it suffers drawbacks compared to LMs that were pre-trained on fewer languages. Wu et al. showed that pairing subsets of closely related languages leads to LMs that outperform mBERT on the respective target language for parsing tasks and Named Entity Recognition [43].

### 2.1.4 On the Availability of Non-English Textual Data

While most available public textual data is written in English, there are multiple projects that provide access to corpora written in other languages, most notably Wikipedia, Common Crawl, and the Leipzig Corpora Collection.

#### 2.1.4.1 Multilingual Text Corpora

The pre-training of modern non-English and multilingual models depend upon vast amounts of available data which in turn require a considerable effort to collect. There exist various large-scale corpora covering a multitude of languages, facilitating the step of data gathering for building language models.

The most well-known collection of multilingual textual data is Wikipedia, spanning 318 languages with an average of 169 316 and a median of 10 646 articles per language at the time of writing (December 2022). The English Wikipedia contains the highest number of articles with 6 125 199 while the smallest Wikipedia corpus with 159 articles is the Wikipedia written in Cree, the language spoken by the North American indigenous people of the same name[2]. Single Wikipedia corpora can be downloaded as raw data dumps[3] and pre-processed with APIs such as WikiExtractor [44].

One of the largest public text repositories available is the one created by the Common Crawl organisation[4]. It consists of textual data crawled from the web that has been collected since 2011. The current version of the repository has a size of several petabytes. While being an impressively large dataset, it has a major disadvantage in that it is not subdivided by language. As such, it is not convenient to use for building monolingual language models without thorough pre-processing.

The **O**pen **S**uper-large **C**rawled **A**LMAnaCH co**R**pus (OSCAR) [38] addresses the aforementioned shortcoming of the Common Crawl repository by pre-processing and classifying it by language. The current version supports 166 languages.[5] The

[2] https://www.thecanadianencyclopedia.ca/en/article/cree
[3] https://dumps.wikimedia.org/
[4] https://commoncrawl.org/
[5] https://oscar-project.org/post/oscar-v22-01/

language with the largest subset is English with a size of 3.2TB or nearly 377 billion words; the smallest is the Quechua subset with a size of merely 744 bytes or 14 words, Quechua being the most widespread indigenous language family in South America[6].

Another important repository for multilingual web-crawled data is the Leipzig Corpora Collection [45][7]. The repository includes textual data for 293 languages. It consists of crawled and pre-processed data from various sources including news articles and Wikipedia.

#### 2.1.4.2 Low-Resource Languages and Data Augmentation

As discussed in Section 1.2.3, languages such as Luxembourgish face a considerable challenge regarding the creation of well-performing language models: the scarcity of available data. While large-scale repositories for multiple languages are useful, there are many languages that benefit only to a degree from these corpora due to the limited availability of public texts. These languages are known as low-resource languages [46] The lack of resources can be due to various factors such as the low number of speakers or the language being a spoken language rather than a written one.

As mentioned in Section 2.1.3.2, BERT-like language models need a huge amount of data in order to perform well. BERT, CamemBERT, GottBERT, and BETO were trained on 13GB, 135GB, 145GB, and 17GB, respectively, consisting of hundreds of millions of sentences. This amount of data is usually not available for low-resource languages. Indeed, while there are Luxembourgish portions of Wikipedia and the OSCAR corpus available, they contain merely 38MB and 16MB of textual data, respectively. The Luxembourgish portion of the Leipzig Corpora Collection is considerably larger, consisting of 305MB of textual data. However, this amount is still several orders of magnitude smaller than the dataset used to pre-train the original BERT model.

Due to the challenges posed by the Luxembourgish language involving the presence of foreign words and spelling variation, building reliable models may prove difficult as shown by numerous failed language technologies for other low resource languages [47]. The difficulties may stem from sociolinguistic variations where the vocabulary used by target user base differs from the one used by the model [48], or the high number of loanwords and frequent code switching [10]. It is important to keep these pitfalls in mind when building datasets and models for languages such as Luxembourgish.

In order to mitigate the lack of data, new data can be artificially created using Data Augmentation. The term *Data Augmentation* is used to describe a set of techniques that are used to generate synthetic data samples by slightly modifying authentic data. Originating from the field of Computer Vision and Image Classification, where techniques such as rotations, cropping, or random noise injection are used to automatically enhance datasets and improve model performances, data augmentation has also proved useful in the field of NLP. Numerous simple text altering techniques have been proposed to create synthetic textual data that can be used for supervised as well as unsupervised NLP tasks and language model pre-training.

[6] `https://www.omniglot.com/writing/quechua.htm`

[7] `https://wortschatz.uni-leipzig.de/de/download`

Kobayashi et al. [49] 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 et al. [50] leveraged synonym replacement, random insertion, random swap, and random deletion to further increase the performance of text classifiers. Liu et al. [51] 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, yet they are useful to improve the performance of trained models.

## 2.2 Common NLP Tasks

There is a broad range of supervised NLP tasks, varying greatly in hardness and complexity. They range from simple parsing tasks, over text classification tasks to NP-hard NLU tasks. Figure 2.4 shows an overview of a selection of NLP tasks ordered by difficulty. [4] The performance of LMs and other architectures is typically evaluated by how well they perform in downstream NLP tasks. In this section, we discuss some of the NLP tasks that will be investigated throughout this thesis and provide examples for each relevant task. We divide these tasks into three categories: sequence-to-sequence tasks, text classification, and Natural Language Understanding tasks.

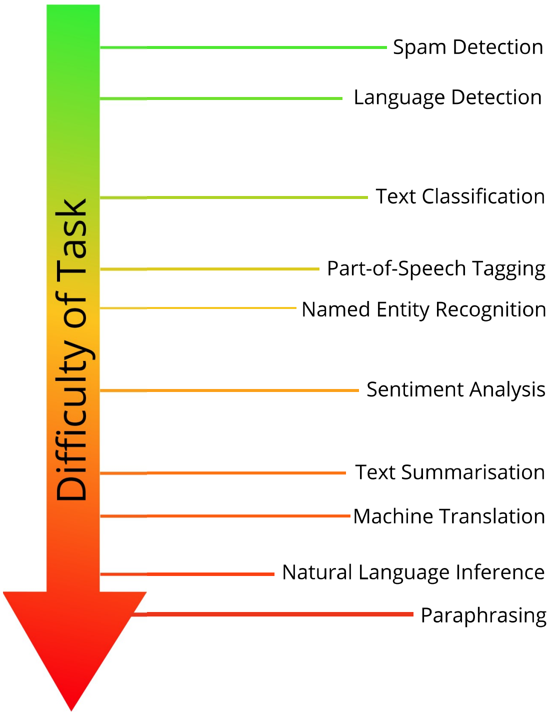


Figure 2.4: Selection of NLP tasks ordered by difficulty (adapted from Vajjala et al. [4])

## 2.2.1 Sequence-to-Sequence Tasks

Sequence-to-Sequence (seq2seq) tasks take sentences as input and assign labels to each word in the sentence, i.e. they map sequences of words to sequences of labels. In this work, we look at two well-known seq2seq tasks: Part-of-Speech (POS) Tagging (cf. Chapter 6 and 7) and Named Entity Recognition (NER) (cf. Chapters 3, 6, and 7)

### Part-of-Speech Tagging

POS-Tagging is a common parsing task, the objective of which is to attribute grammatical classes to each word in a given sentence, e.g. *noun*, *adjective*, *determiner*, etc. A sentence could be tagged using the Penn Treebank tagset [52] as follows:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| He | owes | a | lot | of | money | to | the | bank |
| PP | VBP | DT | NN | IN | NN | IN | DT | NN |

While most of POS tags are the same across languages, a lot of languages have unique grammatical classes. For instance, there is an *APPRART* tag for both Luxembourgish and German, which denotes a contraction of a preposition and a determiner.

### Named Entity Recognition

Named Entity Recognition is a common Information Extraction task to detect proper names in a text. For simple (or coarse-grained) NER, we typically distinguish between four main types: *person* (PER), *organisation* (ORG), *location* (LOC), and *miscellaneous* (MISC), as well as a fifth type *Other* (O) to denote words that are not named entities. Using these tags, a sentence could be tagged as follows:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Frank | owes | a | lot | of | money | to | BGL BNP Paribas |
| PER | O | O | O | O | O | O | ORG |

These tags can be further extended to include a sizeable number of classes which are more informative than the standard NER tags.

## 2.2.2 Text Classification Tasks

Some of the most common tasks in NLP are text classification (TC) tasks where a given text is assigned a label from a predefined set of classes. TC tasks include Intent Classification (IC) for conversational AI (cf. Chapters 5, 6, and 7) and News Classification (NC) (cf. Chapters 4, 6 and 7).

### Intent Classification

Intent Classification tasks are fundamental tasks needed for conversational AI such as digital personal assistants or chatbots. Given a user query or prompt, the objective is to recognise the underlying intent of the message. Evidently, potential labels for intents are strongly dependent on the domain in which the model is deployed. The most well-known dataset for IC is the Airline Travel Information System (ATIS) dataset [53] containing examples related to commercial flight travel such as:

| | | |
|---|---|---|
| I need a flight tomorrow from Columbus to Minneapolis | → | ask_flight_information |
| What is the distance from Los Angeles International Airport to Los Angeles | → | ask_travel_distance |
| Show me the fares from Dallas to San Francisco | → | ask_airfare |

### News Classification

News Classification typically refers to the categorisation of news articles into predefined categories such as news related to *politics*, *sports*, or celebrities. One of the largest NC datasets is the *News Category Dataset* [54][55], containing nearly 210 000 news headlines from HuffPost[8] divided into 42 categories including *entertainment*, *business*, and *comedy*:

| | | |
|---|---|---|
| 23 Of The Funniest Tweets About Cats And Dogs This Week (Sept. 17-23) | → | comedy |
| Biden Says Queen's Death Left 'Giant Hole' For Royal Family | → | politics |
| Privatization Isn't The Answer To Jackson's Water Crisis | → | environment |

One important related task is Fake News Classification aiming to detect whether a news article is factual, or contains falsified or misleading information. This task is particularly important in the context of political and medical news, as false information prevents people from making informed decisions concerning upcoming election cycles or their own health. The *Politifact Fact Check Dataset* is a useful dataset for this task containing nearly 21 000 texts from PolitiFact[9]:

| | | |
|---|---|---|
| A German doctor discovered the COVID-19 vaccines include graphene oxide or graphene hydroxide [...] | → | false |
| Barack Obama has played over 90 rounds of golf as president. | → | true |
| Republicans Mitt Romney and Paul Ryan support m̈assive cuts in Social Security for future generations.¨ | → | half-true |

## 2.2.3 Natural Language Understanding Tasks

The most difficult tasks for NLP to solve are Natural Language Understanding (NLU). They can be regarded as NP-complete problems with regard to computational complexity [56]. In this thesis, we will focus on tasks from the General Language Understanding Evaluation (GLUE) collection [30] (cf. Chapters 4, 6, and 7).

### 2.2.3.1 The General Language Understanding Evaluation Benchmark

The most well-known benchmark to evaluate the NLU capabilities of a language model is the GLUE benchmark by Wang et al. [30] They are a collection of nine tasks divided into three broad categories: single-sentence tasks, similarity and paraphrasing tasks, and inference tasks. Table 2.1 shows each task in the collection.

### Single-Sentence Tasks

The collection features two single-sentence classification tasks that are relevant for this thesis: Corpus of Linguistic Acceptability (CoLA) (cf. Chapter 4) and the Stanford Sentiment Treebank (SST-2) dataset (cf. Chapter 7)

[8] https://www.huffpost.com/

[9] https://www.politifact.com/

Table 2.1: Tasks featured in the GLUE collection

| Corpus | #Train (k) | #Test (k) | Task | Domain |
|---|---|---|---|---|
| Single-Sentence Tasks | | | | |
| CoLA | 8.5 | 1 | acceptability | misc. |
| SST-2 | 67 | 1.8 | sentiment | movie reviews |
| Similarity and Paraphrasing Tasks | | | | |
| MRPC | 3.7 | 1.7 | paraphrase | news |
| STS-B | 7 | 1.4 | sentence similarity | misc. |
| QQP | 364 | 391 | paraphrasing | social QA questions |
| Inference Tasks | | | | |
| MNLI | 393 | 20 | NLI | misc. |
| QNLI | 105 | 5.4 | QA/NLI | Wikipedia |
| RTE | 2.5 | 3 | NLI | news, Wikipedia |
| WNLI | 0.634 | 0.146 | co-reference/NLI | fiction books |

The CoLA task [57] consists of determining whether or not a given sentence is grammatically sound, which is a useful dataset to train grammar checkers. The samples in the dataset can be grouped together into clusters of sentences that are similar to each other, some of them being grammatically acceptable while others are not acceptable:

| | | |
|---|---|---|
| The book was written by John. | → | acceptable |
| The book was written from John. | → | not_acceptable |
| The book was written by. | → | not_acceptable |

SST-2 [26] is a popular Sentiment Analysis (SA) dataset consisting of determining whether a sentence expresses a positive or a negative sentiment. This dataset is a collection of single sentences from movie reviews such as the following:

| | | |
|---|---|---|
| are more deeply thought through than in most ' right-thinking ' films. | → | positive |
| lend some dignity to a dumb story | → | negative |
| saw how bad this movie was. | → | negative |

### Similarity and Paraphrasing Tasks

While the GLUE benchmark features three similarity and paraphrasing tasks, only one is relevant to this work: the Microsoft Research Paraphrasing Corpus (MRPC) [58] (cf. Chapter 4). Given a sentence pair A and B, the objective is to determine whether or not B expresses the same meaning as A. It is made up of sentence pairs from news articles such as:

| | | |
|---|---|---|
| Yucaipa owned Dominick's before selling the chain to Safeway in 1998 for $2.5 billion.<br>Yucaipa bought Dominick's in 1995 for $693 million and sold it to Safeway for $1.8 billion in 1998. | → | not_paraphrase |
| They had published an advertisement on the Internet on June 10 offering the cargo sale he added.<br>On June 10 the ship's owners had published an advertisement on the Internet offering the explosives for sale. | → | paraphrase |
| A BMI of 25 or above is considered overweight; 30 or above is considered obese.<br>A BMI between 18.5 and 25.9 is considered normal over 25 is considered overweight. | → | not_paraphrase |

### Inference Tasks

There are three Natural Language Inference (NLI) tasks that are relevant to this dissertation: the Multi-Genre NLI (MNLI) task (cf. Chapter 4), the Recognizing Textual Entailment (RTE) task (cf. Chapters 4 and 7), and the Winograd NLI (WNLI) task (cf. Chapters 4, 6, and 7).

All three tasks are sentence pair tasks. Given a pair A and B, the general objective of each task consists of determining whether or not sentence A entails B. However, the tasks and the data sources differ for each task. The data for the MNLI task [59] was collected from a multitude of texts including fiction books, government press releases, and travel guides. It is an inference task with three possible labels: *entailment*, *contradiction*, and *neutral*:

| | | |
|---|---|---|
| At the heart of the sanctuary, a small granite shrine once held the sacred barque of Horus.<br>Horus has a shrine. | → | entailment |
| At the heart of the sanctuary, a small granite shrine once held the sacred barque of Horus.<br>Horus is a god. | → | neutral |
| At the heart of the sanctuary, a small granite shrine once held the sacred barque of Horus.<br>The barque of Horus still remains within a shine inside the sanctuary. | → | contradiction |

The RTE task [60] features data from both news and Wikipedia articles. It is a binary task with the labels *entailment* and *not_entailment*:

| | | |
|---|---|---|
| No Weapons of Mass Destruction Found in Iraq yet.<br>Weapons of Mass Destruction Found in Iraq. | → | not_entailment |
| Lin Piao, after all, was the creator of Mao's "Little Red Book" of quotations.<br>Lin Piao wrote the "Little Red Book". | → | entailment |

The data for WNLI [61] was collected exclusively from fiction books. It is considered the most challenging task in the collection as it combines a co-reference task with an inference task. Given two texts A and B, where A contains one or several pronouns, and B is a substring of A with a pronoun replaced by a noun from A, the task consists of determining whether or not A entails B. The objective essentially is to decide whether or not the pronoun was replaced by the correct noun. Similarly to CoLA, the samples can be clustered into nearly identical sentence pairs, with both correct and incorrect pronoun replacements:

| | | |
|---|---|---|
| John couldn't see the stage with Billy in front of him because he is so short.<br>John is so short. | → | entailment |
| John couldn't see the stage with Billy in front of him because he is so short.<br>Billy is so short. | → | not_entailment |

# PART I

## NLP in the Financial Domain

In the first part, we address the challenges present in the financial domain. In particular, we focus on the handling of names in financial and other sensitive documents. In a first step, we tackle the challenge of recognising names in a document and studying the effect of the domain on the performance of the examined models. In a second step, we study the effect of anonymising personal names on the subsequent processing of a document.

# 3 Evaluating Pre-trained Transformer-based Models on the Task of Fine-Grained Named Entity Recognition

*In this chapter, we compare three Transformer-based models (BERT, RoBERTa, and XLNet) to two non-Transformer-based models (CRF and BiLSTM-CNN-CRF). Furthermore, we apply each model to a multitude of distinct domains. We find that Transformer-based models incrementally outperform the studied non-Transformer-based models in most domains with respect to the F1 score. We also find that the choice of domain significantly influenced the performance regardless of the respective data size.*

This chapter is based on the work published in the following research paper:

- Cedric Lothritz, Kevin Allix, Lisa Veiber, Tegawendé F. Bissyandé, Jacques Klein, Evaluating pretrained transformer-based models on the task of fine-grained named entity recognition, *Proceedings of the 28th International Conference on Computational Linguistics*, 2020

## Contents

## 3.1 Overview

Named Entity Recognition (NER) is part of the fundamental tasks in Natural Language Processing (NLP). The main objective of NER is to detect and classify proper names (named entities) in a free text. Typically, named entities can be subdivided into four broad categories: **persons**, i.e., first and last names, **locations** such as countries or landscapes, **organisations** such as companies or political parties, and **miscellaneous entities** which serves as a catch-all category for other named entities such as brands, meals, or social events. NER is an active research field and state-of-the-art solutions such as spaCy[1], flair [62], and Primer[2] manage to achieve near-human performance. However, classical NER (which we refer to as coarse-grained NER in this chapter) models typically distinguish between only a small number of entity types, usually fewer than a dozen distinct categories.

While this kind of broad classification is sufficient for many applications, there are industrial use-cases in which more precise information is necessary such as financial documents processing in the banking and finance context. For instance, application forms for a business loan are usually supplied with several supporting textual documents. These can contain the names of different types of persons, such as the owner or the CEO of the applying company, the contact person(s) at the issuing bank, finance analysts, or lawyers. The same is true for organisation names such as the name of the issuing bank, a government agency, or the name of the applying company or third-party companies. It is necessary to not only detect entity names, but to also *qualify* and differentiate between various entity types. Indeed, in many contexts the actual name of an entity is important only if it can be associated to a *role*, or any other relevant *quality*. In the banking and finance world for example, the strict regulatory requirements cannot be satisfied with just a list of *who* is involved; knowing *how* entities are involved is a necessity. Furthermore, this kind of preprocessing step could help with compliance and background checks of applicants and affiliated companies. For instance, automatically extracting names from a document and cross-checking them with databases can speed up the process to find blacklisted, fraudulent, or otherwise undesirable companies in the paper work.

The term "Fine-Grained Named Entity Recognition" (FG-NER) was first coined by Fleischman et al. [63]. It describes a subtask of NER, where the objective remains the same as coarse-grained NER, but where the number of entity types is considerably higher. In extreme cases, FG-NER models such as the **FI**ne-**G**rained **E**ntity **R**ecognizer (FIGER) [64] are able to distinguish between more than 100 distinct labels.

Conditional Random Field (CRF) models [65] have been popular for numerous sequence-to-sequence tasks such as NER. They perform reasonably well and can serve as a baseline for the task of FG-NER.

In a previous study, Mai et al. [66] compared the performance of several FG-NER approaches for the English and Japanese languages. They found that the BiLSTM-CNN-CRF model devised by Ma et al. [67] combined with gazetteers performed the

[1] https://spacy.io

[2] https://primer.ai/blog/a-new-state-of-the-art-for-named-entity-recognition/

best in terms of F1 score for the English language. They also found that BiLSTM-CNN-CRF performed well without the use of gazetteers. In fact, among the models that did not make use of gazetteers, BiLSTM-CNN-CRF achieved the highest F1 score.

The introduction of the Transformer [2] and BERT models[3] led to state-of-the-art results in numerous NLP tasks, including NER. Indeed, Devlin et al. reported an F1 score of 92.8% when fine-tuned on the CoNLL-2003 dataset for NER [68], achieving similar results as state-of-the-art models such as Contextual String Embeddings [62] and ELMo Embeddings [69].

One model that improved on the BERT approach is the RoBERTa [32] model by vastly increasing the size of the pre-training dataset and tweaking the learning algorithm. They report that fine-tuned models derived from RoBERTa either matched or improved on BERT models in terms of performance, although they did not perform experiments specifically on the NER task.

Finally, the XLNet model developed by Yang et al. [31] addressed shortcomings of BERT regarding the pre-training approach. Yang et al. [31] reports that XLNet outperforms BERT in 20 NLP tasks such as text classification and language understanding, but they do not report any results on sequence-to-sequence tasks like NER.

While BERT, RoBERTa, and XLNet (which we collectively refer to as Transformer-based models throughout this chapter) achieve state-of-the-art performances in numerous Natural Language Understanding (NLU) tasks, we observe a lack of research in the area of FG-NER. In this chapter, we present an empirical study of the performance of FG-NER approaches derived from a pre-trained BERT, a pre-trained RoBERTa, and a pre-trained XLNet model as well as a comparison to a simple CRF model and the model presented by Ma et al. [67]. Furthermore, we apply these approaches to a large number of distinct domains, with varying numbers of data samples and entity categories.

Our contributions are two-fold:

(a) An empirical study on the performance of Transformer-based approaches on the FG-NER task

(b) An analysis of the impact of the choice of domain on the performance of models trained on FG-NER

We use the EWNERTC dataset published by Sahin et al. [70], containing roughly 7 million data samples in 49 different domains. To the best of our knowledge, our study was the first aiming to precisely evaluate the performance of these existing approaches on the FG-NER task.

The rest of this chapter is organised as follows: Section 3.2 describes our experimental setup for this study, the EWNERTC dataset and the models that we investigate. In Section 3.3, we present the results of our experiments, and answer the research questions that we laid out. Section 3.5 lists various possible shortcomings and limits of our study. Finally, we summarise our findings in Section 3.6.

## 3.2 Experimental Setup

In this section, we enumerate our research questions, present the dataset used in this study and we introduce the different models that we compare against each other.

### 3.2.1 Research Questions

1. RQ1: *Do Transformer-based models outperform the state-of-the-art model for the FG-NER task?* We consider two non-Transformer approach vs. three Transformer-based approaches to determine if there is a noticeable trend.

2. RQ2: *What are the strengths, weaknesses, and trade-offs of each investigated model?* We examine the results of each model with regard to precision, recall, and F1 score to establish the advantages and disadvantages of each approach.

3. RQ3: *How does the choice of the domain influence the performance of the models?* We fine-tune each approach on 49 separate domains and examine differences in the results to determine if the choice of domain has a noticeable impact on the performance of a fine-tuned model.

### 3.2.2 Dataset

For this study, we apply the selected models to the English Wikipedia Named Entity Recognition and Text Categorization (EWNERTC) dataset[3] published by Sahin et al. [71]. It is a collection of automatically categorised and annotated sentences from Wikipedia articles. The original dataset consists of roughly 7 million annotated sentences, divided into 49 separate domains. These 49 domains vary significantly in overall size and number of entity types. The *physics* domain is the smallest subset with 68 sentences, 144 entities and merely 6 distinct entity types. In contrast, the *location* domain is the largest subset with 443 646 sentences, 1 472 198 entities, and 1603 types. Table 3.1 contains statistics for the subsets investigated in this study.

*Physics*, *fashion*, *finance*, *exhibitions*, and *meteorology* are the five smallest sets, consisting of fewer than 3000 sentences each. The largest sets are *government*, *film*, *music*, *people*, and *location* with more than 300 000 sentences each. It is noteworthy that the *physics* dataset is an obvious outlier in terms of size (since the second smallest dataset is the *fashion* dataset, which contains an order of magnitude more sentences). It is possible that the size of the *physics* subset is too small to produce meaningful results.

For this study, the number of entity types was drastically reduced. This measure was taken for two reasons: most entity types appear only a few times in any given subset. Furthermore, the training time for CRF models tends to explode when dealing with a high number of entity types according to Mai et al. [66]. We limited the number of entity types per domain to the top 50 and, if necessary, added a *miscellaneous* type as a catch-all for all remaining named entities.

### 3.2.3 Approaches

In this section, we present the five models that we investigate for this study in more detail and we specify the configuration of each model.

[3] https://data.mendeley.com/datasets/cdcztymf4k/1

Table 3.1: Statistics for the datasets used for this study

| **ID** | **domain** | **#sentences** | **#words** | **#named entities before removal** | **#entity types after removal** | **#entity types** |
|---|---|---|---|---|---|---|
| 1 | physics | 68 | 1916 | 144 | 6 | 5 |
| 2 | fashion | 1043 | 27598 | 2182 | 68 | 20 |
| 3 | finance | 1723 | 42834 | 4121 | 75 | 24 |
| 4 | exhibitions | 1829 | 40162 | 2950 | 131 | 34 |
| 5 | meteorology | 2838 | 69551 | 7659 | 92 | 32 |
| 6 | interests | 3462 | 87318 | 6132 | 150 | 44 |
| 7 | measurement_unit | 3864 | 103222 | 9675 | 103 | 50 |
| 8 | internet | 3915 | 100110 | 9140 | 171 | 50 |
| 9 | engineering | 4475 | 118061 | 11548 | 242 | 50 |
| 10 | chemistry | 4883 | 109289 | 10007 | 100 | 43 |
| 11 | astronomy | 8298 | 201407 | 25072 | 214 | 50 |
| 12 | automotive | 10349 | 270043 | 25364 | 90 | 39 |
| 13 | soccer | 11398 | 280920 | 36620 | 216 | 50 |
| 14 | opera | 11559 | 290749 | 35459 | 227 | 50 |
| 15 | law | 11813 | 320329 | 38737 | 310 | 51 |
| 16 | visual_art | 12059 | 306650 | 32693 | 292 | 51 |
| 17 | basketball | 12604 | 308931 | 51532 | 316 | 50 |
| 18 | computer | 12955 | 321726 | 34494 | 297 | 50 |
| 19 | theater | 15340 | 372118 | 51848 | 408 | 50 |
| 20 | symbols | 21171 | 531455 | 61097 | 728 | 50 |
| 21 | comic_books | 21262 | 541396 | 53764 | 317 | 50 |
| 22 | language | 21306 | 551685 | 62830 | 214 | 50 |
| 23 | religion | 27977 | 721771 | 77446 | 401 | 50 |
| 24 | time | 28903 | 758863 | 84560 | 573 | 50 |
| 25 | royalty | 30587 | 788890 | 110169 | 427 | 50 |
| 26 | games | 31420 | 806552 | 109941 | 433 | 50 |
| 27 | aviation | 36924 | 939383 | 125823 | 344 | 50 |
| 28 | medicine | 37729 | 940578 | 82854 | 402 | 51 |
| 29 | fictional_universe | 39781 | 1010777 | 89567 | 567 | 50 |
| 30 | food | 41160 | 1034233 | 100445 | 415 | 50 |
| 31 | media_common | 49714 | 1269641 | 142084 | 959 | 50 |
| 32 | biology | 53042 | 1248434 | 131518 | 246 | 50 |
| 33 | travel | 59965 | 1467691 | 152712 | 750 | 50 |
| 34 | business | 68244 | 1688935 | 182306 | 1009 | 50 |
| 35 | architecture | 76322 | 1947451 | 237468 | 633 | 50 |
| 36 | geography | 94712 | 2355917 | 296847 | 359 | 51 |
| 37 | military | 95809 | 2548948 | 303750 | 864 | 50 |
| 38 | transportation | 111864 | 2867706 | 403018 | 482 | 50 |
| 39 | award | 117280 | 2733563 | 595158 | 1323 | 51 |
| 40 | book | 135865 | 3351012 | 437189 | 1032 | 50 |
| 41 | organization | 146583 | 3679927 | 502582 | 1215 | 50 |
| 42 | tv | 154152 | 3720607 | 574619 | 935 | 51 |
| 43 | sports | 171645 | 4300796 | 622688 | 961 | 51 |
| 44 | education | 212423 | 5113452 | 783132 | 792 | 51 |
| 45 | government | 331720 | 8380706 | 1170947 | 1182 | 51 |
| 46 | film | 430693 | 9557747 | 1720973 | 1134 | 51 |
| 47 | music | 441220 | 10116628 | 1684479 | 918 | 50 |
| 48 | people | 442683 | 11452451 | 1762255 | 1825 | 50 |
| 49 | location | 443646 | 12525545 | 1472198 | 1603 | 50 |

#### 3.2.3.1 CRF

As CRF models remain largely popular solutions for sequence-to-sequence tasks, we use a simple CRF model as a baseline. We use a large number of context and word shape features such as casing information and whether or not the word contains numerical characters. While simple CRF models generally perform well for coarse-grained NER, they require custom-made features and their usefulness is limited for FG-NER according to Mai et al. [66] who observed that CRF models tend to require too much time to finish when handling a large number of labels. We use the sklearn_crfsuite API[4] for python with the following hyperparameters for training: gradient descent using the L-BFGS method as the training algorithm with a maximum of 100 iterations. The coefficients for L1 and L2 regularisation are fixed to $C_1 = 0.4$ and $C_2 = 0.0$. We use the following features: the word itself, casing information, is the word alphabetical, numerical or alphanumerical, suffixes and prefixes, as well as the words and features in a two-words context window. Considering that the datasets are numerous and very diverse, we decided against using specialised gazetteers/dictionaries for this study, despite their proven usefulness in earlier studies [66].

#### 3.2.3.2 BiLSTM-CNN-CRF

As our state-of-the-art model, we use the implementation of Reimers et al. [72][5] of the BiLSTM-CNN-CRF model proposed by Ma et al. [67]. The model consists of a combination of a convolutional neural network (CNN) layer, a bidirectional long short-term memory (BiLSTM) layer, and a CRF layer. In a first step, the CNN is used to extract character-level representations of given words which are then concatenated with word embeddings to create word level representations of the input tokens. These representations are fed into a forward and a backward LSTM layer, creating a bidirectional encoding of the input sequence. Finally, a CRF layer decodes the resulting representations into the most probable label sequence [67]. Mai et al. [66] achieved the best performance with a combination of gazetteers and BiLSTM+CNN+CRF, but as was mentioned above, we do not use gazetteers for this study due to the diverse nature of our datasets. We use the hyperparameters recommended by Reimers et al. [73] as they were shown to be useful for coarse-grained NER. We also use **Glo**bal **Ve**ctors (GLoVe)[6] word embeddings with 300 dimensions for the same reason.

#### 3.2.3.3 BERT

For our first Transformer-based language model, we use the English *BERT Base* model (cf. Section 2.1.3.2) that we fine-tune on each dataset separately. As we compare models for FG-NER, we chose the cased model as recommended, in order to preserve casing information. We use the Transformers library[7] provided by Huggingface [74] which allows to pre-train and fine-tune BERT models with a simplified procedure using CLI commands.

The *BERT Base* model contains 12 Transformer blocks, 768 hidden layers, 12 self-

[4] https://github.com/TeamHG-Memex/sklearn-crfsuite

[5] https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf

[6] https://github.com/stanfordnlp/GloVe

[7] https://github.com/huggingface/transformers

attention blocks, and 110 million parameters in total. While the *BERT Large* model yields better results in every task that Devlin et al.[3] investigated, the *BERT Base* model can be useful for determining a lower boundary for the performance. Devlin et al.[3] report that the recommended hyperparameters vary depending on the NLP task, but generally the best performances are observed for a batch size in $\{16, 32\}$, a learning rate in {2e-5, 3e-5, 5e-5}, and training epochs in $\{2, 3, 4\}$. After testing on three specific domains (*comic books*, *symbols*, and *fictional universe* with 21 262, 21 171 and 39 781 sentences respectively), we found that a batch size of 16, a learning rate of 5e-5, and 5 training epochs yielded the highest F1 scores.

#### 3.2.3.4 RoBERTa

As our second Transformers-based model, we use the *RoBERTa Base* model (cf. Section 2.1.3.3), which contains 12 Transformer blocks, 768 hidden layers, 12 self-attention heads, and 125 million trainable parameters. We fine-tune it on each dataset separately. Similar to the pre-trained BERT model, the pre-trained RoBERTa model is also cased, making it appropriate for fine-tuning on NER tasks. Liu et al. [32] trained RoBERTa using the same hyperparameters as BERT, except for the number of training epochs which they fixed to ten. We perform a similar grid search as for BERT, i.e., a batch size in $\{16, 32\}$, and a learning rate in $\{2e-5, 3e-5, 5e-5\}$, but training epochs in $\{2, 4, 6, 8, 10\}$. Testing on the *comic books*, *symbols*, and *fictional universe*, we found that a batch size of 16 , a learning rate of 5e-5, and 10 training epochs performed best with regard to F1 score.

#### 3.2.3.5 XLNet

The final model used for this study is the XLNet model by Yang et al. [31] (cf. Section 2.1.3.3). For the comparison, we use the cased *XLNet Base* model with 12 Transformer blocks, 768 hidden layers, 12 self-attention heads, and 110 million parameters. Yang et al. [31] fine-tuned their pre-trained model using the same hyperparameters as the BERT models to compare their performances. We perform the same hyperparameter grid search as for BERT, and get the best F1 score with a batch size of 16, a learning rate of 5e-5 and 5 training epochs for the domains *comic books*, *symbols*, and *fictional universe*.

## 3.3 Experimental Results

In this section, we will answer the three research questions that we formulated for this study (cf. Section 3.2.1). Table 3.2 shows the performance of the five models for each domain. In order to account for the imbalanced distribution of the entity types, we opt to calculate micro-averaged performance scores which takes into account the frequency of every entity type. To facilitate reading, we highlight (in **bold**) the highest F1 score for each domain.

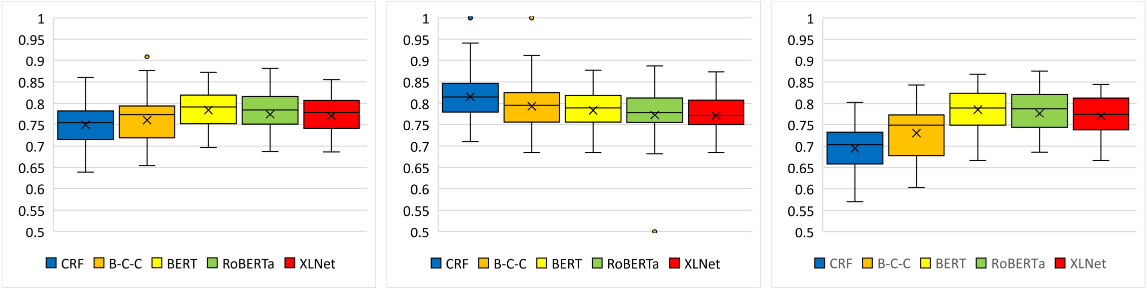
### 3.3.1 RQ1: Do Transformer-based models outperform the state-of-the-art model for the FG-NER task?

The results indicate that, overall, the Transformer-based models outperform CRF and BiLSTM-CNN-CRF in most domains in terms of F1 score. Specifically, the results show that the BERT and RoBERTa models yield the highest and second-highest F1 scores for almost every domain. BERT has the highest F1 score in 36 out of 49 domains, while RoBERTa achieves the best F1 score in 10 out of 49 domains. While

XLNet outperforms BiLSTM-CNN-CRF in most domains, its performance scores are slightly lower than the ones of both the BERT and RoBERTa models. It is also noteworthy that XLNet performs consistently worse than BiLSTM-CNN-CRF in the ten smallest domains.

Figure 3.1a provides the boxplots showing the distributions of the F1 scores over all the domains across the five models. We can make two observations. The boxplots indicate that, on average, all of the Transformer-based models achieve higher performances than both CRF and BiLSTM-CNN-CRF. Furthermore, we can observe that the ranges, and, more importantly, the interquartile ranges of the Transformer-based models are smaller. This indicates that their performances are more stable and less sensitive to the choice of domain than the performances of CRF and BiLSTM-CNN-CRF.

> **RQ1 Answer:** Transformer-based models generally outperform both the CRF and the BiLSTM-CNN-CRF models in terms of F1 score, with BERT yielding the highest results overall. In addition, their performance is also more stable across domains.



(a) Distribution of F1 scores (b) Distribution of Precision (c) Distribution of Recall

Figure 3.1: Distribution the performance of the five models used

### 3.3.2 RQ2: What are the strengths, weaknesses, and trade-offs of each investigated model?

While the Transformer-based models clearly outperform the other models with regard to the F1 score, it is worth examining the precision and recall scores as well. Regarding the precision, the CRF model almost consistently outperforms all of the other models as shown in Table 3.2. When compared to the BiLSTM-CNN-CRF model, the Transformer-based models perform worse in most domains in terms of precision. In fact, BERT outperforms BiLSTM-CNN-CRF in less than half of the domains, RoBERTa outperforms BiLSTM-CNN-CRF in only a third of the domains and XLNet outperforms it in only a fifth of the domains. Figure 3.1b shows the distribution of the precision scores over all the domains across the five models. The boxplots confirm the strength of CRF over the other models. Furthermore, they show that BiLSTM-CNN-CRF performs slightly better than the Transformer-based models, albeit at a loss of stability as indicated by the large range.

On the other hand, the Transformer-based models significantly outperform the other models with regard to recall as seen in Table 3.2. In fact, both BERT and RoBERTa significantly outperform CRF and BiLSTM-CNN-CRF in almost every domain, while

Table 3.2: Micro-averaged results of each model for every domain. Bold text indicates the highest F1 score for the domain.

| | | | CRF | | | LSTM-CNN-CRF | | | BERT | | | RoBERTa | | | XLNet | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | domain | #sents | prec | rec | F1 | prec | rec | F1 | prec | rec | F1 | prec | rec | F1 | prec | rec | F1 |
| 1 | physics | 68 | 1 | 0.778 | 0.875 | 1 | 0.833 | **0.909** | 0.857 | 0.667 | 0.75 | 0.5 | 0.444 | 0.471 | 0.706 | 0.667 | 0.686 |
| 2 | fashion | 1043 | 0.92 | 0.765 | **0.836** | 0.894 | 0.776 | 0.831 | 0.849 | 0.801 | 0.824 | 0.816 | 0.816 | 0.816 | 0.825 | 0.77 | 0.797 |
| 3 | finance | 1723 | 0.859 | 0.708 | 0.776 | 0.83 | 0.731 | 0.777 | 0.807 | 0.796 | 0.802 | 0.794 | 0.839 | **0.815** | 0.768 | 0.759 | 0.764 |
| 4 | exhibitions | 1829 | 0.901 | 0.737 | **0.811** | 0.831 | 0.744 | 0.785 | 0.765 | 0.754 | 0.76 | 0.788 | 0.782 | 0.785 | 0.759 | 0.74 | 0.75 |
| 5 | meteorology | 2838 | 0.748 | 0.675 | 0.709 | 0.75 | 0.753 | 0.751 | 0.746 | 0.79 | 0.767 | 0.755 | 0.792 | **0.773** | 0.722 | 0.742 | 0.732 |
| 6 | interests | 3462 | 0.943 | 0.811 | 0.872 | 0.912 | 0.843 | 0.876 | 0.877 | 0.868 | 0.872 | 0.887 | 0.875 | **0.881** | 0.873 | 0.838 | 0.855 |
| 7 | measurement unit | 3864 | 0.822 | 0.707 | 0.76 | 0.812 | 0.772 | 0.791 | 0.794 | 0.806 | **0.8** | 0.79 | 0.795 | 0.792 | 0.773 | 0.785 | 0.779 |
| 8 | internet | 3915 | 0.83 | 0.63 | 0.716 | 0.768 | 0.657 | 0.709 | 0.727 | 0.712 | 0.719 | 0.749 | 0.725 | **0.737** | 0.73 | 0.687 | 0.708 |
| 9 | engineering | 4475 | 0.856 | 0.63 | 0.726 | 0.764 | 0.691 | 0.726 | 0.734 | 0.722 | 0.728 | 0.739 | 0.725 | **0.732** | 0.694 | 0.689 | 0.691 |
| 10 | chemistry | 4883 | 0.869 | 0.736 | 0.797 | 0.874 | 0.768 | 0.818 | 0.836 | 0.823 | **0.829** | 0.815 | 0.823 | 0.819 | 0.81 | 0.805 | 0.808 |
| 11 | astronomy | 8298 | 0.85 | 0.743 | 0.792 | 0.825 | 0.781 | 0.802 | 0.825 | 0.833 | 0.829 | 0.831 | 0.831 | **0.831** | 0.821 | 0.814 | 0.817 |
| 12 | automotive | 10 349 | 0.799 | 0.735 | 0.766 | 0.788 | 0.779 | 0.784 | 0.792 | 0.816 | **0.803** | 0.773 | 0.797 | 0.785 | 0.772 | 0.801 | 0.786 |
| 13 | soccer | 11 398 | 0.766 | 0.647 | 0.702 | 0.779 | 0.681 | 0.727 | 0.77 | 0.773 | **0.772** | 0.756 | 0.769 | 0.763 | 0.761 | 0.764 | 0.763 |
| 14 | opera | 11 559 | 0.865 | 0.74 | 0.798 | 0.825 | 0.776 | 0.8 | 0.827 | 0.847 | **0.837** | 0.83 | 0.839 | 0.834 | 0.814 | 0.824 | 0.819 |
| 15 | law | 11 813 | 0.792 | 0.64 | 0.708 | 0.756 | 0.701 | 0.727 | 0.75 | 0.759 | 0.754 | 0.758 | 0.752 | **0.755** | 0.761 | 0.745 | 0.753 |
| 16 | visual art | 12 059 | 0.861 | 0.649 | 0.74 | 0.81 | 0.674 | 0.736 | 0.766 | 0.725 | 0.745 | 0.774 | 0.721 | **0.747** | 0.761 | 0.718 | 0.738 |
| 17 | basketball | 12 604 | 0.836 | 0.796 | 0.815 | 0.832 | 0.83 | 0.831 | 0.833 | 0.849 | **0.841** | 0.828 | 0.85 | 0.839 | 0.824 | 0.844 | 0.834 |
| 18 | computer | 12 955 | 0.814 | 0.673 | 0.737 | 0.768 | 0.74 | 0.754 | 0.762 | 0.773 | **0.767** | 0.755 | 0.767 | 0.761 | 0.748 | 0.757 | 0.752 |
| 19 | theater | 15 340 | 0.79 | 0.608 | 0.688 | 0.733 | 0.658 | 0.694 | 0.709 | 0.719 | 0.714 | 0.719 | 0.725 | **0.722** | 0.7 | 0.697 | 0.698 |
| 20 | symbols | 21 171 | 0.72 | 0.571 | 0.637 | 0.715 | 0.62 | 0.664 | 0.723 | 0.727 | **0.725** | 0.724 | 0.712 | 0.718 | 0.711 | 0.699 | 0.705 |
| 21 | comic books | 21 262 | 0.854 | 0.711 | 0.776 | 0.808 | 0.749 | 0.777 | 0.808 | 0.829 | 0.818 | 0.818 | 0.821 | **0.82** | 0.796 | 0.815 | 0.805 |
| 22 | language | 21 306 | 0.803 | 0.74 | 0.77 | 0.79 | 0.764 | 0.777 | 0.81 | 0.816 | **0.813** | 0.799 | 0.809 | 0.804 | 0.787 | 0.8 | 0.793 |
| 23 | religion | 27 977 | 0.805 | 0.697 | 0.747 | 0.787 | 0.761 | 0.774 | 0.808 | 0.81 | **0.809** | 0.8 | 0.796 | 0.798 | 0.787 | 0.791 | 0.789 |
| 24 | time | 28 903 | 0.717 | 0.565 | 0.632 | 0.697 | 0.63 | 0.662 | 0.716 | 0.722 | **0.719** | 0.704 | 0.704 | 0.704 | 0.704 | 0.705 | 0.705 |
| 25 | royalty | 30 587 | 0.804 | 0.725 | 0.762 | 0.785 | 0.76 | 0.772 | 0.786 | 0.798 | **0.792** | 0.779 | 0.788 | 0.784 | 0.774 | 0.785 | 0.779 |
| 26 | games | 31 420 | 0.839 | 0.741 | 0.787 | 0.796 | 0.77 | 0.783 | 0.79 | 0.813 | **0.801** | 0.789 | 0.81 | 0.799 | 0.768 | 0.791 | 0.779 |
| 27 | aviation | 36 924 | 0.795 | 0.712 | 0.751 | 0.779 | 0.73 | 0.754 | 0.789 | 0.807 | **0.798** | 0.781 | 0.797 | 0.789 | 0.774 | 0.79 | 0.782 |
| 28 | medicine | 37 729 | 0.848 | 0.697 | 0.765 | 0.797 | 0.755 | 0.776 | 0.802 | 0.788 | **0.795** | 0.791 | 0.788 | 0.789 | 0.799 | 0.791 | **0.795** |
| 29 | fictional universe | 39 781 | 0.874 | 0.756 | 0.811 | 0.845 | 0.781 | 0.812 | 0.843 | 0.855 | **0.849** | 0.841 | 0.848 | 0.845 | 0.837 | 0.842 | 0.839 |
| 30 | food | 41 160 | 0.801 | 0.648 | 0.717 | 0.746 | 0.69 | 0.717 | 0.776 | 0.788 | **0.782** | 0.76 | 0.766 | 0.763 | 0.752 | 0.774 | 0.763 |
| 31 | media common | 49 714 | 0.862 | 0.723 | 0.786 | 0.819 | 0.755 | 0.786 | 0.806 | 0.825 | **0.815** | 0.807 | 0.819 | 0.813 | 0.803 | 0.812 | 0.807 |
| 32 | biology | 53 042 | 0.854 | 0.771 | 0.811 | 0.843 | 0.807 | 0.825 | 0.834 | 0.847 | **0.84** | 0.832 | 0.837 | 0.834 | 0.836 | 0.837 | 0.836 |
| 33 | travel | 59 965 | 0.822 | 0.696 | 0.754 | 0.803 | 0.719 | 0.759 | 0.784 | 0.79 | **0.787** | 0.764 | 0.772 | 0.768 | 0.779 | 0.777 | 0.778 |
| 34 | business | 68 244 | 0.803 | 0.634 | 0.709 | 0.756 | 0.666 | 0.708 | 0.765 | 0.771 | **0.768** | 0.755 | 0.759 | 0.757 | 0.752 | 0.754 | 0.753 |
| 35 | architecture | 76 322 | 0.709 | 0.588 | 0.643 | 0.685 | 0.627 | 0.654 | 0.707 | 0.722 | **0.715** | 0.688 | 0.701 | 0.694 | 0.685 | 0.695 | 0.69 |
| 36 | geography | 94 712 | 0.813 | 0.728 | 0.768 | 0.801 | 0.752 | 0.776 | 0.798 | 0.815 | **0.806** | 0.795 | 0.804 | 0.799 | 0.789 | 0.799 | 0.794 |
| 37 | military | 95 809 | 0.836 | 0.731 | 0.78 | 0.82 | 0.778 | 0.798 | 0.816 | 0.827 | **0.821** | 0.811 | 0.821 | 0.816 | 0.809 | 0.823 | 0.816 |
| 38 | transportation | 111 864 | 0.828 | 0.738 | 0.781 | 0.834 | 0.804 | 0.819 | 0.845 | 0.857 | **0.851** | 0.845 | 0.85 | 0.848 | 0.839 | 0.844 | 0.841 |
| 39 | award | 117 280 | 0.702 | 0.617 | 0.657 | 0.702 | 0.671 | 0.686 | 0.685 | 0.716 | **0.7** | 0.682 | 0.707 | 0.694 | 0.689 | 0.703 | 0.695 |
| 40 | book | 135 865 | 0.761 | 0.604 | 0.675 | 0.717 | 0.639 | 0.676 | 0.711 | 0.73 | **0.721** | 0.708 | 0.723 | 0.716 | 0.716 | 0.722 | 0.719 |
| 41 | organization | 146 583 | 0.769 | 0.64 | 0.698 | 0.765 | 0.674 | 0.717 | 0.767 | 0.776 | **0.771** | 0.756 | 0.766 | 0.761 | 0.762 | 0.768 | 0.765 |
| 42 | tv | 154 152 | 0.725 | 0.574 | 0.641 | 0.733 | 0.603 | 0.662 | 0.697 | 0.696 | **0.696** | 0.688 | 0.686 | 0.687 | 0.702 | 0.684 | 0.693 |
| 43 | sports | 171 645 | 0.781 | 0.705 | 0.741 | 0.799 | 0.767 | 0.783 | 0.806 | 0.822 | **0.814** | 0.801 | 0.816 | 0.808 | 0.807 | 0.819 | 0.813 |
| 44 | education | 212 423 | 0.734 | 0.653 | 0.691 | 0.747 | 0.706 | 0.726 | 0.769 | 0.78 | **0.774** | 0.763 | 0.774 | 0.769 | 0.769 | 0.774 | 0.771 |
| 45 | government | 331 720 | 0.81 | 0.725 | 0.765 | 0.815 | 0.764 | 0.789 | 0.821 | 0.828 | **0.825** | 0.816 | 0.824 | 0.82 | 0.824 | 0.825 | 0.824 |
| 46 | film | 478 479 | 0.75 | 0.68 | 0.713 | 0.743 | 0.695 | 0.718 | 0.769 | 0.773 | **0.771** | 0.766 | 0.767 | 0.766 | 0.772 | 0.768 | 0.77 |
| 47 | music | 462 949 | 0.786 | 0.654 | 0.714 | 0.78 | 0.668 | 0.72 | 0.744 | 0.744 | **0.744** | 0.739 | 0.736 | 0.737 | 0.752 | 0.736 | **0.744** |
| 48 | people | 442 683 | 0.836 | 0.771 | 0.802 | 0.847 | 0.795 | 0.82 | 0.83 | 0.83 | **0.83** | 0.825 | 0.821 | 0.823 | 0.834 | 0.825 | 0.829 |
| 49 | location | 443 646 | 0.809 | 0.703 | 0.752 | 0.8 | 0.713 | 0.754 | 0.79 | 0.789 | **0.79** | 0.775 | 0.772 | 0.774 | 0.784 | 0.775 | 0.78 |

XLNet outperforms them in most. The same result can be observed in Figure 3.1c. The Transformer-based models not only outperform the other models, but their interquartile ranges are significantly smaller as well. This difference in recall score also explains the higher F1 scores for the Transformer-based models.

> **RQ2 Answer:** CRF shows its strength in terms of precision, BERT, RoBERTa, and XLNet perform well with regard to both recall and F1 score, with BERT usually achieving the highest performances. The BiLSTM-CNN-CRF model acts as a trade-off between CRF and the Transformer-based models.

### 3.3.3 RQ3: How does the choice of the domain influence the performance of the models?

Figure 3.1a shows that while different models may achieve significantly different performance, no approach yields a significant breakthrough, with regard to the others, for the task at hand, and all leave room for improvement. The five tested models obtained relatively stable performances, as is visible from the fact that boxes, which represent the performance measurements of 50% of the domains, cover only a ±0.05 band around the average.

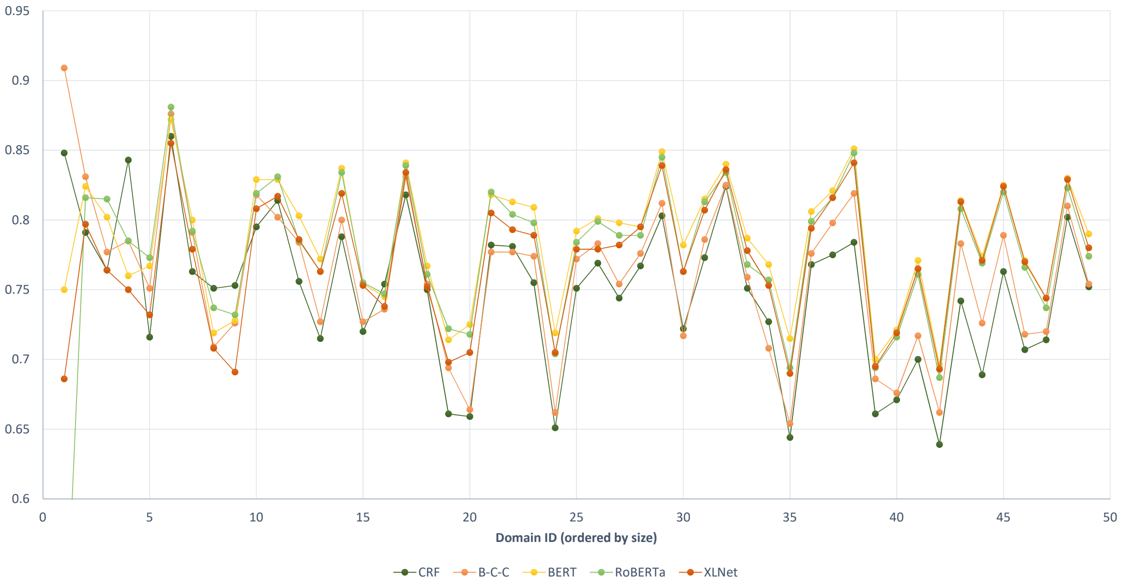


Figure 3.2: Performances of the five models for every domain (in terms of F1 score).

Figure 3.2, that plots the F1 scores for every domain (ordered by size), reveals however that all models are similarly impacted by domains: with the exceptions of the four smallest domains (left-most on Figure 3.2), when one model achieves a lower performance than its overall average, all models are also performing worse than their overall averages. We also note that the per-domain variations in performance cannot be explained by the size of the domains (since the performance looks erratic across all domain sizes). The variations in performance can also not be explained by the perceived obscurity of the domain. Intuitively, it is conceivable that the scientific domains (*physics*, *engineering*, *chemistry*, *biology*), as well as *finance* and *law* could lead to worse performances as they typically contain terms that are rarely encountered during the pre-training phase of the respective word embeddings or language models. However, there is no clear indication that this is the case as only the *engineering* and *law* domains consistently lead to results that are significantly below the median performance of the respective model, while the other domains lead to results that are either close to or significantly above the median performance.

Overall, the results are a clear indication that most domains are either: (a) *relatively hard* for every model, or (b) *relatively easy* for every model. This suggests that no model manages to acquire a massively better language *understanding* that would make it able to avoid the difficulties faced by the other models, at least in the context of FG-NER.

Furthermore, the ranking of the five models is very stable across domains: given the fact that one specific model performs the best (resp. the worst) for one domain, it can reliably be predicted that this model will also perform the best (resp. the worst) across all domains. It follows that some models do bring a sometime incremental, but nonetheless measurable improvement over other models. Nevertheless, we note that for the four smallest domains, the difference in performance from one model to another is more important, and no ranking pattern is visible.

The performance variations between domains that we see in our results have also been reported in the study by Guo et al. [75], who investigated the stability of coarse-grained NER across domains for the Chinese language. Notably, when trained on the *sports* domain, their baseline has a significantly higher F1 score than the other domains. The same is true here, but it has to be noted that they use the classic NER-labels, i.e., *person*, *location*, *organisation*, and *miscellaneous*, rather than domain-specific labels.

> **RQ3 Answer:** We observe significant discrepancies when applying the models to different domains. Moreover, when a model is performing better (resp. worse) on one domain, the other models also perform better (resp. worse). This suggests that while Transformer-based models can indeed bring significant performance improvements, their language *understanding* may not be outstandingly different. Indeed, if they were clearly different, we could have reasonably expected to note different patterns in the performance for the FG-NER task (i.e., they would not systematically perform well/badly for the same domains).

## 3.4 Related Work

### 3.4.1 Fine-Grained Named Entity Recognition

Early efforts to develop a fine-grained approach to NER were made by Bechet et al. [76], where they focused on differentiating between first names, last names, countries, towns, and organisations. While this would be considered coarse-grained by today's standards, they do split the classical NER labels *person* and *location* into more nuanced labels. FG-NER was first described as "fine grained classification of named entities" by Fleischman et al. [63]. They focused on a fine-grained label set for personal names, dividing the generic *person* label into eight subcategories, i.e., *athlete*, *politician/government*, *clergy*, *businessperson*, *entertainer/artist*, *lawyer*, *doctor/scientist*, and *police*. They experimented with a variety of classic machine learning approaches for this task, and achieved promising results of 68.1%, 69.5%, and 70.4% in terms of accuracy for SVM, a feed-forward neural network, and a C4.5 decision tree, respectively. Furthermore, Ling et al. [64] introduced their fine-grained entity recognizer (FIGER), which can distinguish between 112 different labels and handle multi-label classification. Mai et al. [66] presented an empirical study on FG-NER prior to the rise of Transformer-based models (which are the focus of our study). They targeted an English dataset containing 19 800 sentences and a Japanese dataset which contained 19 594 sentences, dividing the named entities into 200 categories. They compared performances for FIGER, BiLSTM-CNN-CRF, and a hierarchical CRF+SVM classifier, which classifies an entity into a coarse-grained category before further classifying it into a fine-grained subcategory. Furthermore, they combine some of the aforementioned methods with gazetteers and category embeddings to further improve the performance of the models. They found that the BiLSTM-CNN-CRF model by Ma et al. [67] combined with gazetteer information performed the best for the English language with an F1 score of 83.14% while BiLSTM-CNN-CRF with both gazetteers and category embeddings yielded an F1 score of 82.29%, and 80.93% without either gazetteers or category embeddings.

## 3.5 Threats to Validity

This study was conducted on the EWNERTC dataset [70] which was annotated automatically. We are operating under the assumption that the annotations are accurate. However, while Sahin et al.[71] conducted an evaluation for the Turkish counterpart of the dataset (TWNERTC), they did not evaluate the English one. Nevertheless, EWNERTC is the largest publicly available dataset that we could find and that is relevant for FG-NER studies. We further proposed to reduce the potential noise in labelling by considering only the subset associated to top labels (cf. Section 3.2.2).

Performance measurements can be impacted by sub-optimal implementation of algorithms. To mitigate this threat, we collected the models' implementations that were released by their original authors, and already leveraged in previous studies, and we reused them in the settings they were designed for.

While we conducted grid searches to determine optimised hyperparameters for the CRF, BERT, RoBERTa and XLNet models, we did not specifically optimise the hyperparameters for the the BiLSTM-CNN-CRF model due to the induced computational costs. Furthermore, as pointed out in Section 3.2.3.1, due to the large number of domains, we decided against using gazetteers even though they would likely have increased the F1 scores of the non-Transformer-based models.

## 3.6 Summary

In this chapter, we presented an empirical study of the performance of various Transformer-based models for the FG-NER task on a multitude of domains and compared them to both CRF and BiLSTM-CNN-CRF models (which are commonly used in the literature for the NER task).

We concluded that while the Transformer-based models did not manage to outperform non-Transformer-based models in terms of precision, we observed a consistent increase in recall and F1 scores in most domains. We noticed, however, significant differences in performance for a selection of domains that could not be explained by the size of the respective datasets. This study yields the main insight that while Transformer-based models can indeed bring significant performance improvements, they do not necessarily revolutionise the achievements in FG-NER to the same extent they did in other NLP tasks.

# 4 Evaluating the Impact of Text De-Identification on Downstream NLP Tasks

*In this chapter, we investigate the impact of text de-identification on the performance of nine downstream NLP tasks. We focus on the anonymisation and pseudonymisation of personal names and compare six different anonymisation strategies for two state-of-the-art pre-trained models. Based on these experiments, we formulate recommendations on how the de-identification should be performed to guarantee accurate NLP models.*

This chapter is based on the work published in the following research paper:

- Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Tegawendé F. Bissyandé, Jacques Klein, Andrey Boytsov, Clément Lefebvre, and Anne Goujon. Evaluating the Impact of Text De-Identification on Downstream NLP Tasks, Nordic Conference on Computational Linguistics, 2023

## Contents

## 4.1 Overview

Protection of personal data has been a hot topic for decades [77]. Careless sharing of data between and within companies, cyber attacks, and other forms of data breaches can lead to catastrophic leaks of confidential data, potentially resulting in invasion of people's privacy and identity theft.

To mitigate damages and to hold bad actors accountable, many countries introduced various laws that aim to protect confidential data. Examples of such legislation include the Health Insurance Portability and Accountability Act (HIPAA) for healthcare confidentiality [78], and the Gramm–Leach–Bliley Act (GLBA) in the financial domain [79]. Most notably, with the introduction of the General Data Protection Regulation (GDPR), protection of any personally identifiable information was codified into EU law in 2018 [9]. Failure to comply with these regulations can lead to huge fines in case of a data breach. Indeed, the amount of fines for GDPR violations adds up to over 1.5 trillion euros with the largest single fine of 746 million euros being imposed on Amazon.[1]

In order to mitigate data leaks and avoid costly fines, organisations such as financial institutes and hospitals are required to anonymise or pseudonymise sensitive data before processing them further. Similarly, automated NLP models should ideally be trained using de-identified data as resulting models could potentially violate a number of GDPR guidelines such as the individuals' right to be forgotten, and the right to explanation. Furthermore, models can be manipulated to partially recreate the training data [80], which can result in disastrous data breaches. On the other hand, however, de-identification of texts can lead to loss of information and meaning, making NLP models trained on de-identified data less reliable as a result [81]. Intuitively, this in turn could lead to a decrease in performance of such models when compared to models trained on non-anonymised text. As such, it is crucial to choose an appropriate anonymisation strategy to mitigate this loss of information and avoid performance drops of trained models.

In this study, we investigate the impact of text de-identification on the performance of downstream NLP tasks, focusing on the anonymisation and pseudonymisation of personal names only. This allows us to select from a wide array of NLP tasks as most datasets contain a large number of personal names whereas other types of names are less commonly found. Specifically, we compare six different anonymisation strategies, and two Transformer-based pre-trained model architectures in our experiments: the popular BERT [3] architecture and the state-of-the-art ERNIE [33] architecture. Furthermore, we look into nine different NLP tasks of varying degrees of difficulty. In addition, we set out to do a limited qualitative analysis of the performance and shortcomings of each fine-tuned model using the Language Interpretability Tool [82].

Our contributions are two-fold:

(a) An empirical study on the impact of popular anonymisation strategies on numerous downstream NLP tasks

[1] at the time of writing this work, according to `https://www.privacyaffairs.com/gdpr-fines/`

(b) A superficial qualitative analysis to discover patterns in the decision making of our models

## 4.2 Experimental Setup

In this section, we enumerate our research questions, present the datasets used in this study and we introduce the different anonymisation strategies that we compare against each other. We also briefly introduce the pre-trained models we use.

### 4.2.1 Research Questions

1. RQ1: *Which anonymisation strategy is the most appropriate for downstream NLP tasks?* We evaluate and rank the performance of models trained on de-identified data to determine which anonymisation strategy has the lowest negative impact on a model.

2. RQ2: *Should a model be trained on original or de-identified data?* We train pairs of models on original and on de-identified data, and determine which model performs better on a de-identified test set.

3. RQ3: *In what cases do certain models fine-tuned on de-identified data fail?* Using a popular Explainability approach (LIME[83]), we investigate to what degree the de-identification changed the decision of a given model for a selection of samples.

### 4.2.2 Datasets

Table 4.1: Statistics for the datasets. Size of datasets, number of names found in the training set (#names), number of unique names found in the training set (#unique), percentage of samples that contains at least one name (i.e. the percentage of samples to be de-identified) (%de-identified), and the type of the classification task (binary/multiclass)

| dataset | FND | NBD | FED | MRPC | RTE | WNLI | CoLA | MNLI | ETC |
|---|---|---|---|---|---|---|---|---|---|
| train set | 4382 | 1374 | 8980 | 3668 | 2489 | 635 | 6039 | 39 999 | 6354 |
| dev set | 690 | 196 | 997 | 407 | 276 | 71 | 851 | 5000 | 926 |
| test set | 1237 | 395 | 1926 | 1725 | 800 | 146 | 1661 | 5396 | 1798 |
| #names | 68 890 | 15 610 | 30 404 | 3324 | 3685 | 898 | 2600 | 85 999 | 6550 |
| #unique | 7500 | 3247 | 6104 | 1729 | 2042 | 102 | 335 | 10 460 | 2807 |
| %de-identified | 90.9 | 83.9 | 55.7 | 43.1 | 51 | 61.9 | 41 | 93.8 | 42.6 |
| type | binary | multi | binary | binary | binary | binary | binary | multi | multi |

For this study, we selected several downstream tasks that greatly vary in complexity, ranging from simple text classification to complicated Natural Language Understanding (NLU) tasks featured in the GLUE benchmark collection [30] (cf. Section 2.2.3.1). We ensured that each set contains a considerable number of personal names. Table 4.1 shows a breakdown and statistics for each dataset. These datasets are detailed hereafter. We release the original as well as the de-identified datasets for most tasks.[2]

#### 4.2.2.1 Fake News Detection

Disinformation is a prevalent problem, especially in the areas of medicine and politics. In order to combat the spreading of false information and maintain the trust in

[2] `https://github.com/lothritz/anonymisation_paper`

the free press, several actors are engaged in reliably recognising fake news. The Fake News Detection (FND)[3] task is a binary classification task which consists of determining whether a given news article is real or contains false information. The dataset consists of 6309 political news articles, 50% of which are classified as fake. As it is mainly made up of political news articles, it contains a large number of personal names. The training set includes 68 890 names. Unsurprisingly, more than 90% of the news articles contain personal names.

#### 4.2.2.2 News Bias Detection

Expanding on the FND task, Bharadwaj et al. [84] created and annotated a corpus for a more fine-grained, multiclass news classification task. It serves for detecting problematic or unreliable news articles and features the following labels: bs(bullshit), (political) bias, junk science, conspiracy theory, state(-controlled news), hate(-speech), and satire. We removed some articles from the dataset as they were not in English. After removal, the dataset is made up of 1965 news articles. Similarly to the FND dataset, the number of personal names is relatively high with 15 610 names, and 83.9% of the articles including at least one name.

#### 4.2.2.3 Fraudulent Email Detection

Fraudulent and spam emails can be a dangerous threat to personal data security and lead to identity theft, making the automated detection of fraudulent emails a crucial task. For this study, we use the Fraud Email dataset created byRadev et al. [85]. It is a collection of 11 903 emails, 57% of which are legitimate, and 43% are spam or otherwise fraudulent. The dataset contains 30 404 names in the training set. Considering the large number of names, the number of emails with at least one name is comparably low with only 55%.

#### 4.2.2.4 Microsoft Research Paraphrase Corpus

Dolan et al. [58] created the Microsoft Research Paraphrase Corpus (MRPC). Being part of the GLUE benchmark collection, it is an important dataset for evaluating the capabilities of modern language models. As the name suggests, the task consists of determining whether or not two given texts are paraphrases of each other. Given a text pair A and B, the corresponding label is 1 if text B is a paraphrase of text A, and 0 otherwise. The MRPC set contains 5625 sentence pairs, 66% of which are paraphrase pairs, while 34% are not. The training set contains 3324 names, with 43.1% of the sentence pairs containing at least one name.

#### 4.2.2.5 Recognizing Textual Entailment

Similar to the MRPC dataset, the Recognizing Textual Entailment (RTE) dataset [60] is part of the GLUE collection. This task consists of determining whether a claim can be logically inferred from a given premise, known as linguistic entailment. Given a text pair A and B, the corresponding label is 1 if text A entails text B, and 0 otherwise. The set contains 3563 sentence pairs, 51% of which are examples of linguistic entailment, and 49% of which are not. There are 3685 names in the training set, with 51% of the text pairs containing at least one name.

[3] The dataset can be found at: `https://www.kaggle.com/shubh0799/fake-news`

#### 4.2.2.6 Winograd Natural Language Inference

The Winograd Natural Language Inference (WNLI) dataset was introduced by Levesque et al. [61] and is part of the GLUE collection. The dataset consists of text pairs A and B, where A contains one or several pronouns. Text B is a substring of A with one of the pronouns replaced by a word or name. The task consists of determining whether or not A entails B. Depending on the sentence and the anonymisation strategy, ground truths can be accidentally falsified, making this task's evaluation problematic. The dataset contains 853 sentence pairs, with 46% entailments, and 54% non-entailments. The training set contains 898 names, and 61.9% of its sentence pairs contain names.

#### 4.2.2.7 Corpus of Linguistic Acceptability

The Corpus of Linguistic Acceptability (CoLA) was introduced by Warstadt et al. [57] and is part of the GLUE collection. The task consists of determining whether or not a given sentence makes grammatical sense. As the test set is not labelled, we used the provided training set to construct a new training, validation, and test set. The final dataset contains 8551 samples, 70% of which are grammatically correct sentences. The number of personal names is comparably low at 2600 names, and merely 41% of the sentences containing at least one name.

#### 4.2.2.8 Multigenre Natural Language Inference

Similarly to most datasets used for this study, the Multigenre Natural Language Inference (MNLI) [59] is part of the GLUE collection. Similarly to WNLI, it is a sentence-pair inference task. Given sentences A and B, the task consists of determining whether A entails, contradicts, or is independent from B. Due to the size of the dataset and the low number of sentences containing names, we reduced the dataset by nearly 90%, resulting in 50 395 sentence pairs. 35% of the samples are labeled *entailment*, 33% are *contradiction*, and 32% are *neutral.* As we removed mostly sentences without names, our resulting dataset has the highest percentage of sentence pairs containing at least one name with 93.8%. It also contains the highest absolute number of names with 85 999.

#### 4.2.2.9 Email Topic Classification (Proprietary)

The Email Topic Classification Dataset (ETC) is related to the financial domain and was provided by our partners at BGL BNP Paribas. As such, it is a proprietary dataset consisting of sensitive emails from clients, and thus cannot be publicly released. However, it serves as an authentic use-case for our study. The task consists of classifying the given emails along 19 broad topics related to banking activities such as *credit cards*, *wire transfers*, *account management* etc., which will then be forwarded to the appropriate department. We selected a subset of the provided dataset, such that each topic is represented equally. More specifically, for each topic in the set, we randomly selected $\simeq$ 500 emails, for a total of nearly 9000 emails. Furthermore, the dataset is multilingual, but we perform our experiments on the emails written in French due to the high number of samples.

### 4.2.3 Anonymisation Strategies

In this section, we present the six anonymisation strategies that we consider for this study. These strategies are commonly found in the literature [86, 87]. They largely fall into three categories: replacement by a generic token, removal of names, and replacement by a random name. Table 4.2 shows the differences between each strategy on a simple example.

Table 4.2: Example for each anonymisation strategy

| Original | "Hi, this is Paul, am I speaking to John?" | "Sorry, no, this is George. John is not here today." |
|---|---|---|
| AS1 | "Hi, this is ENTNAME, am I speaking to ENTNAME?" | "Sorry, no, this is ENTNAME. ENTNAME is not here today." |
| AS2 | "Hi, this is ENTNAME1, am I speaking to ENTNAME2?" | "Sorry, no, this is ENTNAME1. ENTNAME2 is not here today." |
| AS3 | "Hi, this is ENTNAME1, am I speaking to ENTNAME2?" | "Sorry, no, this is ENTNAME3. ENTNAME2 is not here today." |
| AS4 | "Hi, this is , am I speaking to " | "Sorry, no, this is . is not here today." |
| AS5 | "Hi, this is Bert, am I speaking to Ernie?" | "Sorry, no, this is Elmo. Kermit is not here today." |
| AS6 | "Hi, this is Jessie, am I speaking to James?" | "Sorry, no, this is Meowth. James is not here today." |

#### 4.2.3.1 AS1: Singular Generic Token

One straightforward de-identification technique is to replace each name by a generic token: *ENTNAME*. This technique prevents any re-identification of anonymised documents. As a downside, it is not possible to distinguish between different people mentioned in a given document, making certain NLU tasks involving Entity Linking such as the WNLI task more difficult. On the other hand, tasks where certain names could introduce bias could potentially perform better when using this strategy.

#### 4.2.3.2 AS2: Unique Generic Token Per Mention

Similarly to AS1, AS2 replaces each name by a generic token. However, rather than using the same token for every name, each name will be replaced by a unique token. Specifically, names get replaced by $ENTNAME+n$ where $n$ increases every time a name is found in a given sentence. Similarly to AS1, some NLU tasks such as the WNLI task will be more difficult as there is no way to link two given de-identified tokens.

#### 4.2.3.3 AS3: Unique Generic Token Per Mention with Identity Mapping

AS3 is almost identical to AS2, with the difference that identical names are mapped to the same de-identified token. Intuitively, this strategy improves on AS2 as it preserves the link between names. As such, we expect downstream tasks that involve Entity Linking to perform better when using AS3 when compared to AS1 and AS2.

#### 4.2.3.4 AS4: Removal of Name

For AS4, rather than replacing names, we instead remove them completely from the dataset. Similarly to AS1, this strategy should reduce bias for tasks such as FND and NBD. However, we would expect AS4 to perform significantly worse in the CoLA task as removing words from a sentence will almost certainly render previously grammatically correct sentences nonsensical. This in turn may produce a high number of false positives in the datasets.

#### 4.2.3.5 AS5: Random Name

Rather than introducing new generic tokens, for AS5, we opt to pseudonymise the datasets by replacing every name with a different name. As the choice of name is random, it is likely that two different names are mapped to the same pseudonym. AS5 has the same downside as AS1 and AS2 in that we expect it to perform worse in tasks such as WNLI.

#### 4.2.3.6 AS6: Random Name with Identity Mapping

Similarly to AS5, AS6 replaces names with random names. However, similarly to AS3, identical names will be mapped to identical pseudonyms, improving on AS5. As such, we expect AS6 to perform better in the WNLI task than AS5.

### 4.2.4 Name Detection

In Chapter 3, we determined that the BERT model is generally best suited for the task of name detection in most domains. Following this result, we fine-tune a *BERT Large* model on the task of Personal Name Detection. We use the CoNLL-2003 dataset for Named Entity Recognition [68] and modify it by relabeling every non-*Person* entity as non-entity. The resulting training set consists of 204 567 words, 11 128 are *Person* entities and 193 439 are labeled as non-entities.[4] The resulting model achieved an F1 score of 0.9694, precision of 0.9786, and a recall of 0.9694 on the modified CoNLL-2003 test set. We use this fine-tuned model to detect and replace names from the training, validation, and test set of the selected downstream tasks.

### 4.2.5 Model Training

We compare the impact of anonymisation strategies using two Transformer-based models: BERT [3] and ERNIE [33]. For our study, we use the transformers library by Huggingface [74] as our framework. Furthermore, we take a grid-search based approach to determine the most appropriate fine-tuning parameters for each downstream task.

#### 4.2.5.1 BERT

Being one of the most important advancements in recent years, BERT models remain popular choices for many NLP applications. (cf. Section 2.1.3.2) For this study, we use the uncased English BERT *Base* model with 12 Transformer blocks, 768 hidden layers, 12 self-attention heads, and 110 million trainable parameters. For the ETC task, we are working on emails written in French. Despite the existence of French BERT models such as CamemBERT [40], we choose to do our experiments with mBERT model as we also use a mulitilingual ERNIE model because there is no French ERNIE model to the best of our knowledge.

#### 4.2.5.2 ERNIE

ERNIE (cf. Section 2.1.3.3) is currently one of the highest ranking language models on the GLUE benchmark leaderboard[5]. Similarly to BERT *Base*, the ERNIE *Base* model has 12 Transformer blocks, 768 hidden layers, and 12 self-attention heads, allowing for a fair comparison between the BERT and ERNIE models. Specifically, we use the ERNIE 2.0 model published by Sun et al. [33] for most tasks. For the ETC task, as already mentioned, we once again use a multilingual model, i.e., the ERNIE-M model published by Ouyang et al. [88].

[4] The dataset used to to train the de-identification model can be found at `https://github.com/lothritz/anonymisation_paper/tree/main/anonymisation_model`

[5] `https://gluebenchmark.com/leaderboard`

### 4.2.6 Fine-tuning Parameters

In order to fine-tune the pre-trained BERT and ERNIE models for the selected tasks, we need to choose appropriate hyperparameters for the batch size, learning rate, and number of training epochs as suggested by Devlin et al. [3] and Sun et al. [33]. We perform a grid search on the original datasets to find the optimised configuration for each task. For BERT models, we perform a grid search with batch size in $\{16, 32\}$, learning rate in {2e-5, 3e-5, 5e-5}, and training epochs in $\{1, 2, 3, 4, 5\}$. For ERNIE models, we choose the same ranges for learning rate and training epochs, but the possible batch sizes are in $\{4, 8, 16, 32, 64, 128, 256, 512\}$. Furthermore, as Sun et al. [33] published appropriate hyperparameters for their ERNIE models on GLUE tasks, we reuse them for the MRPC, RTE, WNLI, CoLA, and MNLI tasks. Table 4.3 shows the final hyperparameters used for fine-tuning.

Table 4.3: Hyperparameters for fine-tuning pre-trained models for downstream tasks

| | BERT | | | ERNIE | | |
|---|---|---|---|---|---|---|
| Task | batch size | learning rate | #epochs | batch size | learning rate | #epochs |
| FND | 16 | 5e-5 | 1 | 8 | $2^{-5}$ | 1 |
| NBD | 16 | 5e-5 | 3 | 8 | $2^{-5}$ | 5 |
| FED | 32 | 3e-5 | 3 | 32 | $5^{-5}$ | 1 |
| MRPC | 16 | 5e-5 | 3 | 32 | $3^{-5}$ | 4 |
| RTE | 16 | 5e-5 | 4 | 4 | $2^{-5}$ | 4 |
| WNLI | 16 | 3e-5 | 4 | 8 | $2^{-5}$ | 4 |
| ColA | 16 | 5e-5 | 3 | 64 | $3^{-5}$ | 3 |
| MNLI | 16 | 5e-5 | 2 | 512 | $3^{-5}$ | 3 |
| ETC | 16 | 5e-5 | 5 | 8 | $3^{-5}$ | 3 |

## 4.3 Experimental Results

Table 4.4: Results of our fine-tuned models. We highlight in green (↑) the models that outperform the models trained on original data, in red (↓) the models that do not.

| | | BERT | | | | | | | ERNIE | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task | Metric | Original | AS1 | AS2 | AS3 | AS4 | AS5 | AS6 | Original | AS1 | AS2 | AS3 | AS4 | AS5 | AS6 |
| FND | F1 | 0.973 | 0.976↑ | 0.974↑ | 0.969↓ | 0.965↓ | 0.968↓ | 0.971↓ | 0.968 | 0.962↓ | 0.960↓ | 0.960↓ | 0.956↓ | 0.956↓ | 0.963↓ |
| NBD | F1 | 0.653 | 0.658↑ | 0.647↓ | 0.654↑ | 0.681↑ | 0.674↑ | 0.683↑ | 0.678 | 0.681↑ | 0.684↑ | 0.695↑ | 0.709↑ | 0.653↓ | 0.669↓ |
| FED | F1 | 0.994 | 0.995↑ | 0.996↑ | 0.996↑ | 0.996↑ | 0.994 | 0.995↑ | 0.996 | 0.994↓ | 0.993↓ | 0.994↓ | 0.993↓ | 0.995↓ | 0.993↓ |
| MRPC | F1 | 0.791 | 0.786↓ | 0.769↓ | 0.768↓ | 0.797↑ | 0.792↑ | 0.783↓ | 0.811 | 0.824↑ | 0.817↑ | 0.799↓ | 0.832↑ | 0.826↑ | 0.82↑ |
| RTE | Acc | 0.691 | 0.67↓ | 0.654↓ | 0.639↓ | 0.624↓ | 0.644↓ | 0.666↓ | 0.703 | 0.696↓ | 0.665↓ | 0.671↓ | 0.683↓ | 0.716↑ | 0.676↓ |
| WNLI | F1 | 0.520 | 0.530↑ | 0.526↑ | 0.551↑ | 0.586↑ | 0.541↑ | 0.535↑ | 0.561 | 0.472↓ | 0.557↓ | 0.564↑ | 0.595↑ | 0.614↑ | 0.550↓ |
| CoLA | MCC | 0.555 | 0.520↓ | 0.522↓ | 0.524↓ | 0.443↓ | 0.495↓ | 0.532↓ | 0.519 | 0.517↓ | 0.543↑ | 0.556↑ | 0.385↓ | 0.540↑ | 0.542↑ |
| MNLI | Acc | 0.754 | 0.742↓ | 0.730↓ | 0.734↓ | 0.745↓ | 0.742↓ | 0.747↓ | 0.789 | 0.774↓ | 0.750↓ | 0.759↓ | 0.770↓ | 0.776↓ | 0.773↓ |
| ETC | F1 | 0.626 | 0.624↓ | 0.683↑ | 0.617↓ | 0.619↓ | 0.616↓ | 0.595↓ | 0.642 | 0.635↓ | 0.696↑ | 0.642 | 0.635↓ | 0.627↓ | 0.621↓ |

In this section, we show and evaluate the results of our experiments and address the research questions introduced in Section 4.2.1. For each task we investigate, and for each pre-trained model, we fine-tune a model on the original dataset and each of our six de-identified datasets. We do five runs for each case, and take the average of the performance of each run. We then compare the average performance for each AS to the performance of the models trained on original data. Table 4.4 shows the average performance of every fine-tuned model. For each of the GLUE tasks, we use the metric recommended by Wang et al. [30]. We use F1 score for the remaining classification tasks.

Table 4.5: Ranking scores for fine-tuned models. **Bold text** indicates the winner according to Borda Count, underlined text indicates the winner according to Instant Runoff.

|  | BERT |  |  |  |  |  | ERNIE |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task | AS1 | AS2 | AS3 | AS4 | AS5 | **AS6** | AS1 | AS2 | AS3 | AS4 | **AS5** | AS6 |
| FND | 5 | 4 | 2 | 0 | 1 | **3** | 4 | 3 | 3 | 1 | **1** | 5 |
| NBD | 2 | 0 | 1 | 4 | 3 | **5** | 2 | 3 | 4 | 5 | **0** | 1 |
| FED | 2 | 5 | 5 | 5 | 0 | **2** | 4 | 2 | 4 | 2 | **5** | 2 |
| MRPC | 3 | 1 | 0 | 5 | 4 | **2** | 3 | 1 | 0 | 5 | **4** | 2 |
| RTE | 5 | 3 | 1 | 0 | 2 | **4** | 4 | 0 | 1 | 3 | **5** | 2 |
| WNLI | 1 | 0 | 4 | 5 | 3 | **2** | 0 | 2 | 3 | 4 | **5** | 1 |
| CoLA | 2 | 3 | 4 | 0 | 1 | **5** | 1 | 4 | 5 | 0 | **2** | 3 |
| MNLI | 3 | 0 | 1 | 4 | 3 | **5** | 4 | 0 | 1 | 2 | **5** | 3 |
| ETC | 4 | 5 | 2 | 3 | 1 | **0** | 3 | 5 | 4 | 3 | **1** | 0 |
| Total | 27 | 21 | 20 | 26 | 18 | **28** | 25 | 20 | 25 | 25 | **28** | 21 |
| Average | 3 | 2.33 | 2.22 | 2.89 | 2 | **3.11** | 2.78 | 2.22 | 2.78 | 2.78 | **3.11** | 2.33 |

Table 4.6: Results of testing the original models on de-identified data. We highlight in green (↑) the models that significantly outperform the matching model in Table 4.4 using a Wilcoxon test, in red (↓) the models that perform significantly worse, in black the models that do not perform significantly differently.

|  |  | BERT |  |  |  |  |  |  | ERNIE |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task | Metric | Original | AS1 | AS2 | AS3 | AS4 | AS5 | AS6 | Original | AS1 | AS2 | AS3 | AS4 | AS5 | AS6 |
| FND | F1 | 0.973 | 0.933↓ | 0.910↓ | 0.907↓ | 0.950↓ | 0.963↓ | 0.963↓ | 0.968 | 0.951↓ | 0.938↓ | 0.935↓ | 0.957↑ | 0.967↑ | 0.967↑ |
| NBD | F1 | 0.653 | 0.566↓ | 0.551↓ | 0.546↓ | 0.601↓ | 0.602↓ | 0.609↓ | 0.678 | 0.683 | 0.684 | 0.659↓ | 0.687↓ | 0.683↑ | 0.683↑ |
| FED | F1 | 0.994 | 0.995 | 0.995 | 0.995 | 0.996 | 0.996 | 0.996 | 0.996 | 0.995 | 0.995 | 0.995 | 0.996 | 0.996 | 0.996 |
| MRPC | F1 | 0.791 | 0.809↑ | 0.811↑ | 0.811↑ | 0.819↑ | 0.816↑ | 0.814↑ | 0.811 | 0.848↑ | 0.848↑ | 0.849↑ | 0.852↑ | 0.804↓ | 0.834↑ |
| RTE | Acc | 0.691 | 0.665↓ | 0.663↑ | 0.669↑ | 0.670↑ | 0.645↑ | 0.660↓ | 0.700 | 0.703↑ | 0.701↑ | 0.693↑ | 0.699↑ | 0.688↓ | 0.704↑ |
| WNLI | F1 | 0.520 | 0.504↓ | 0.504↓ | 0.504↓ | 0.504↓ | 0.504↓ | 0.504↓ | 0.561 | 0.435↓ | 0.442↓ | 0.467↓ | 0.506↓ | 0.458↓ | 0.428↓ |
| CoLA | MCC | 0.555 | 0.376↓ | 0.515↓ | 0.528↑ | 0.335↓ | 0.549↑ | 0.550↑ | 0.519 | 0.427↓ | 0.537↓ | 0.511↓ | 0.313↓ | 0.518↓ | 0.523↓ |
| MNLI | Acc | 0.754 | 0.753↑ | 0.724↓ | 0.753↑ | 0.753↑ | 0.744↑ | 0.744↓ | 0.789 | 0.783↑ | 0.545↓ | 0.760↑ | 0.772↑ | 0.669↓ | 0.765↓ |

### 4.3.1 RQ1: Which anonymisation strategy is the most appropriate for downstream NLP tasks?

In order to determine the most appropriate strategy, we consider two ranking-based approaches: Borda Count and Instant Runoff [89]. For both approaches, we determine the score $s_{a,t}$ for each anonymisation strategy (AS, indexed by $a$) and for each task (indexed by $t$) in the following way: The best of the six approaches gets a score of five, then the second best approach gets a score of four, and so on.

The final *Borda Count* score for a given anonymisation strategy $A$ is defined as $\sum_{t=0}^{T} s_{A,t}$ (where $T$ is the total number of tasks, here, nine). The model with the highest score is considered the best.

*Instant Runoff* is an iterative procedure. For each iteration, we count the number of wins for each AS, where an AS is considered a winner in a given task if its corresponding fine-tuned model outperforms every other model. We then eliminate the AS with the lowest number of wins and update the scores accordingly. We repeat this process until one AS remains, or until we cannot eliminate further ASs.

Table 4.5 shows the scores for each model and the winning anonymisation strategies according to the aforementioned approaches. For BERT models, we see that AS1, AS4, and AS6 are the best performing strategies according to Borda count, AS6 being a close winner. Instant Runoff leads to similar results with AS4 and AS6 reaching the final iteration, and AS6 being the overall winner. Furthermore, we note a lower variance in the scores for AS6 when compared to AS4. In contrast, when evaluating ERNIE models, we note that AS5 models are performing significantly better than every other strategy according to Borda Count. Similarly, AS5 also wins the Instant Runoff with AS4 and AS5 making it to the final round. Overall, it appears that using random names over generic tokens to de-identify textual data is the preferable solution as AS1, AS2, AS3 models, which were all trained on data with generic tokens, usually rank low.

Interestingly, it appears that the intuitive predictions that we made in Section 4.2.3 do not all reflect the outcome of our experiments. As expected, AS4 performs much worse in the CoLA task than all the other models, likely due to the introduction of false positives in the training set. On the other hand, we also assumed AS1, AS2, and AS5 to perform worse for WNLI. While AS1 does perform worse than the original model when trained using the ERNIE architecture, we see similar or better results for AS2, and AS5. We also predicted AS6 to yield higher performances than AS5, but this does not seem to be the case, either. Finally, we predicted that, in general, AS3 and AS6 would perform better than AS2 and AS5, respectively, but in our experiments, we see mixed results.

**RQ1 Answer:** De-identification using random names yielded the best results with AS5 and AS6 reducing performances the least depending on the architecture used.

### 4.3.2 RQ2: Should a model be trained on original or de-identified data?

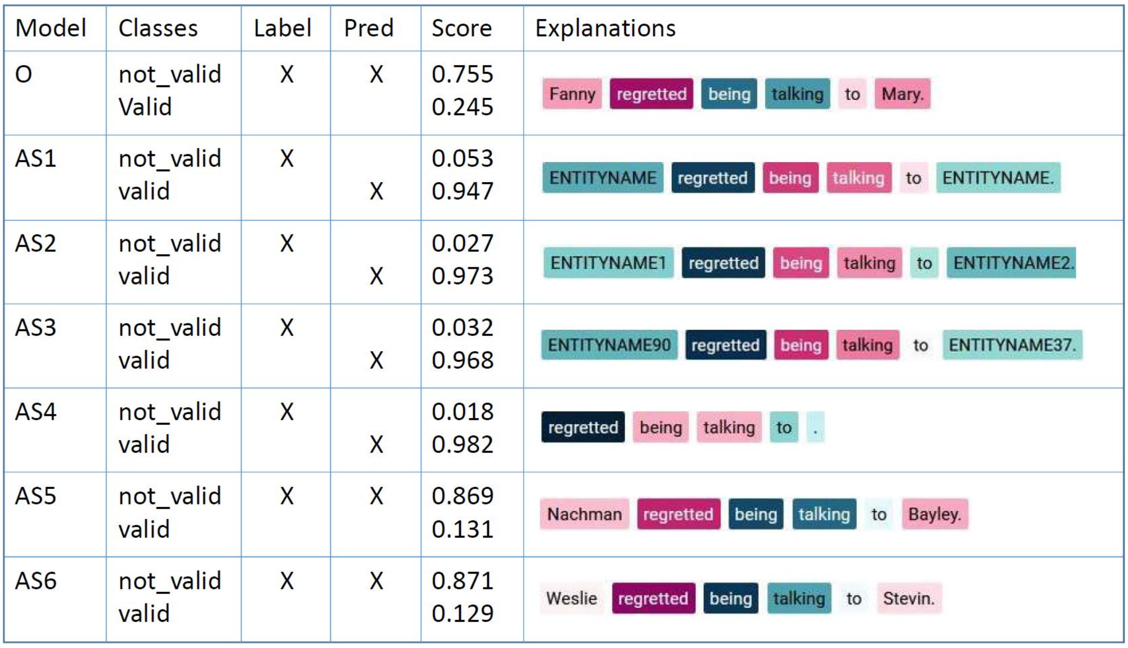
In order to answer this question, we investigate the performance of models trained on original data on the de-identified test sets and compare them to the models trained directly on de-identified data. Table 4.6 shows the results of testing models trained on the original training sets on de-identified test sets. We find that nearly half of the models trained on de-identified data outperform the counterpart model trained on original data. While there is not always a clear trend, we observe that the original models almost consistently perform better in the MRPC and RTE tasks, and perform worse in the WNLI and CoLA tasks, regardless of the architecture used. Furthermore, for BERT models, the models trained on de-identified data consistently perform worse on the FND and NBD tasks. For the ERNIE models, the models trained on original data consistently perform better on the FED task ever so slightly. Despite these observations, we also notice that the performance losses are oftentimes very high, specifically for the NBD, WNLI, and CoLA tasks, while performance gains tend to be lower.

**RQ2 Answer:** This is inconclusive as we notice different trends depending on the investigated task.

### 4.3.3 RQ3: In what cases do certain models fine-tuned on de-identified data fail?

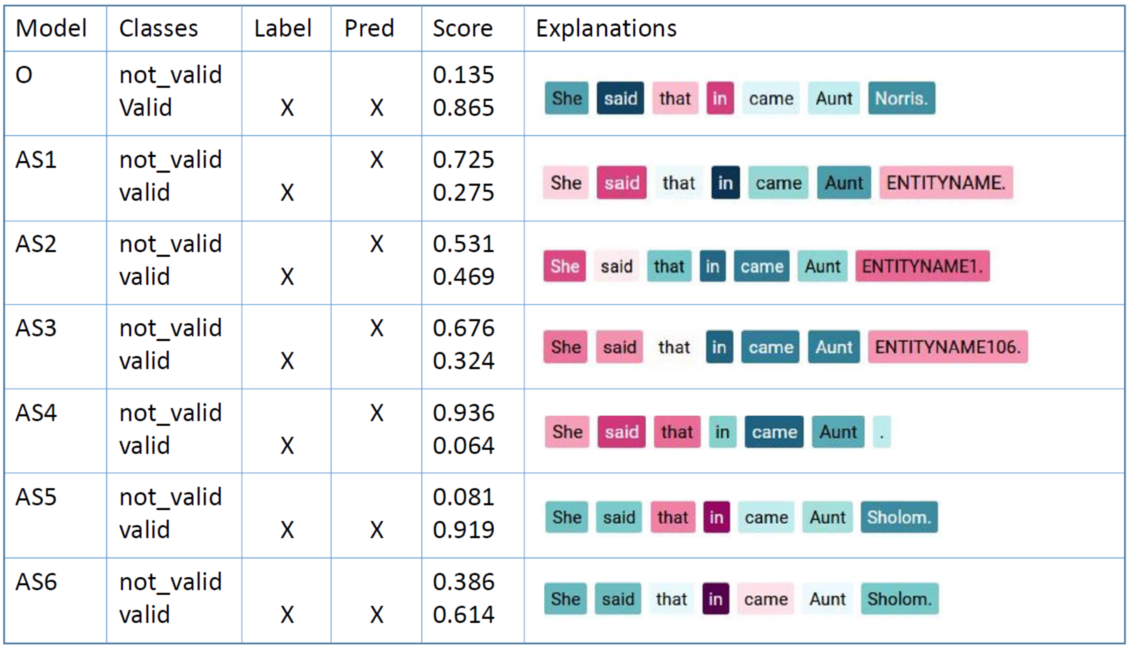
In order to answer this question, we use the Language Interpretability Tool (LIT) [82]. It is a convenient tool that can be used to visualise textual datasets and performances of NLP-models. More importantly, it allows to create explanations of model predictions for given samples using a number of salience maps including Local Interpretable Model-Agnostic Explanations (LIME) [83]. We use the tool to do a limited qualitative analysis of our fine-tuned models. We chiefly investigated examples that were correctly classified by the model trained on original data, but incorrectly on most AS models. Figure 4.1 shows LIME explanations for an example from the CoLA dataset. Tokens that are coloured blue influenced the model towards the prediction it made while red coloured tokens show a negative impact on the model's prediction. A darker shade indicates a higher impact on the prediction. Intuitively, AS4 should be the only anonymisation strategy leading to a misclassification due to the high number of false positives in the training set as mentioned in Section 4.2.3.4. However, the only models that made the correct prediction are the original, AS5, and AS6 models which are the only models that were trained using real names. Furthermore, we observe that each model highlighted the pattern that indicates that the given sentence is not valid, i.e. "being talked". We found several examples where the models trained on generic anonymisation tokens failed to predict the correct class. Even though this observation is not reflected by the overall performance of the models in Table 4.4, it appears that using generic tokens lowers the performance of the fine-tuned model as we observed this pattern several times. Figures 4.2, 4.3, and 4.4 contain further examples of similar patterns appearing for both the CoLA and MNLI tasks.

> **RQ3 Answer:** In the investigated cases, we notice that generic tokens lead to less certainty in the decision of the model.



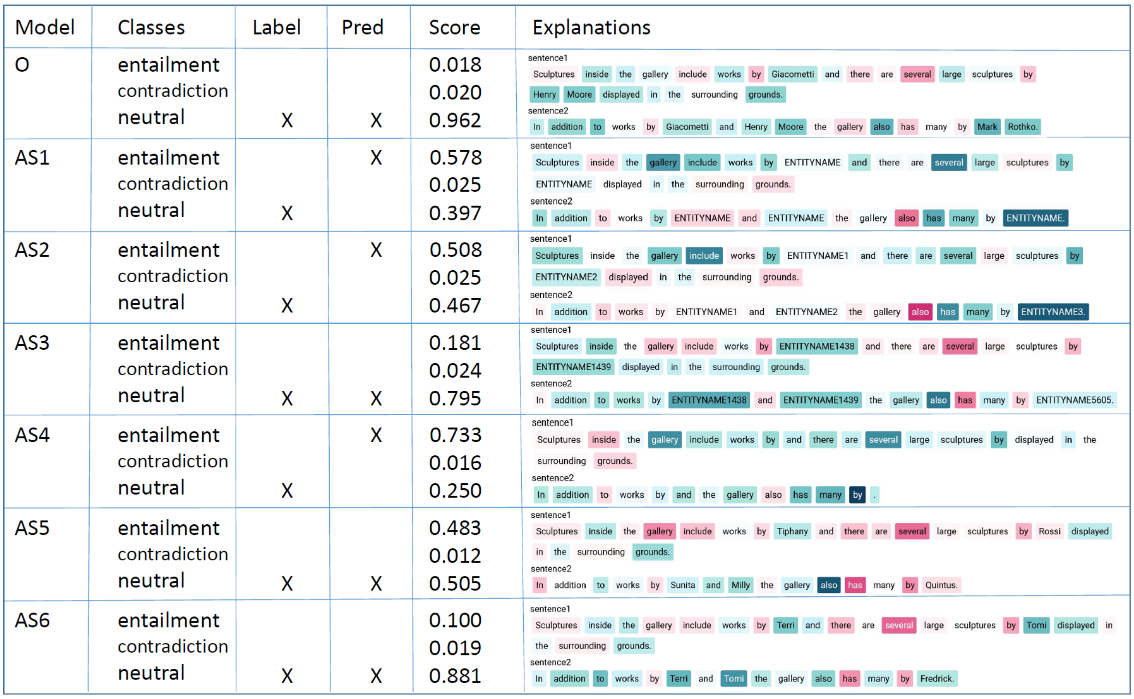
| Model | Classes | Label | Pred | Score | Explanations |
|---|---|---|---|---|---|
| O | not_valid<br>Valid | X | X | 0.755<br>0.245 | Fanny regretted being talking to Mary. |
| AS1 | not_valid<br>valid | X | <br>X | 0.053<br>0.947 | ENTITYNAME regretted being talking to ENTITYNAME. |
| AS2 | not_valid<br>valid | X | <br>X | 0.027<br>0.973 | ENTITYNAME1 regretted being talking to ENTITYNAME2. |
| AS3 | not_valid<br>valid | X | <br>X | 0.032<br>0.968 | ENTITYNAME90 regretted being talking to ENTITYNAME37. |
| AS4 | not_valid<br>valid | X | <br>X | 0.018<br>0.982 | regretted being talking to . |
| AS5 | not_valid<br>valid | X | X | 0.869<br>0.131 | Nachman regretted being talking to Bayley. |
| AS6 | not_valid<br>valid | X | X | 0.871<br>0.129 | Weslie regretted being talking to Stevin. |

Figure 4.1: Predictions and LIME explanations of each model on a given example



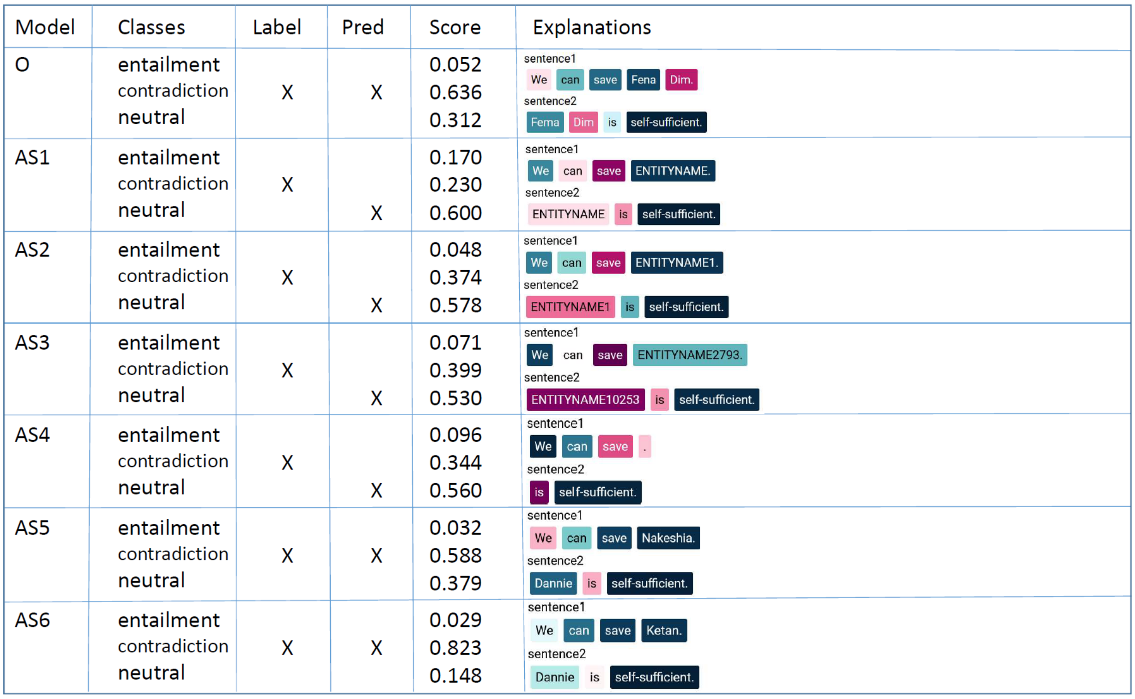
| Model | Classes | Label | Pred | Score | Explanations |
|---|---|---|---|---|---|
| O | not_valid | | | 0.135 | She said that in came Aunt Norris. |
| | Valid | X | X | 0.865 | |
| AS1 | not_valid | | X | 0.725 | She said that in came Aunt ENTITYNAME. |
| | valid | X | | 0.275 | |
| AS2 | not_valid | | X | 0.531 | She said that in came Aunt ENTITYNAME1. |
| | valid | X | | 0.469 | |
| AS3 | not_valid | | X | 0.676 | She said that in came Aunt ENTITYNAME106. |
| | valid | X | | 0.324 | |
| AS4 | not_valid | | X | 0.936 | She said that in came Aunt . |
| | valid | X | | 0.064 | |
| AS5 | not_valid | | | 0.081 | She said that in came Aunt Sholom. |
| | valid | X | X | 0.919 | |
| AS6 | not_valid | | | 0.386 | She said that in came Aunt Sholom. |
| | valid | X | X | 0.614 | |

Figure 4.2: Predictions and LIME explanations of each model on a *valid* example from the CoLA dataset



| Model | Classes | Label | Pred | Score | Explanations |
|---|---|---|---|---|---|
| O | entailment | | | 0.018 | sentence1<br>Sculptures inside the gallery include works by Giacometti and there are several large sculptures by Henry Moore displayed in the surrounding grounds.<br>sentence2<br>In addition to works by Giacometti and Henry Moore the gallery also has many by Mark Rothko. |
| | contradiction | | | 0.020 | |
| | neutral | X | X | 0.962 | |
| AS1 | entailment | | X | 0.578 | sentence1<br>Sculptures inside the gallery include works by ENTITYNAME and there are several large sculptures by ENTITYNAME displayed in the surrounding grounds.<br>sentence2<br>In addition to works by ENTITYNAME and ENTITYNAME the gallery also has many by ENTITYNAME. |
| | contradiction | | | 0.025 | |
| | neutral | X | | 0.397 | |
| AS2 | entailment | | X | 0.508 | sentence1<br>Sculptures inside the gallery include works by ENTITYNAME1 and there are several large sculptures by ENTITYNAME2 displayed in the surrounding grounds.<br>sentence2<br>In addition to works by ENTITYNAME1 and ENTITYNAME2 the gallery also has many by ENTITYNAME3. |
| | contradiction | | | 0.025 | |
| | neutral | X | | 0.467 | |
| AS3 | entailment | | | 0.181 | sentence1<br>Sculptures inside the gallery include works by ENTITYNAME1438 and there are several large sculptures by ENTITYNAME1439 displayed in the surrounding grounds.<br>sentence2<br>In addition to works by ENTITYNAME1438 and ENTITYNAME1439 the gallery also has many by ENTITYNAME5605. |
| | contradiction | | | 0.024 | |
| | neutral | X | X | 0.795 | |
| AS4 | entailment | | X | 0.733 | sentence1<br>Sculptures inside the gallery include works by and there are several large sculptures by displayed in the surrounding grounds.<br>sentence2<br>In addition to works by and the gallery also has many by . |
| | contradiction | | | 0.016 | |
| | neutral | X | | 0.250 | |
| AS5 | entailment | | | 0.483 | sentence1<br>Sculptures inside the gallery include works by Tiphany and there are several large sculptures by Rossi displayed in the surrounding grounds.<br>sentence2<br>In addition to works by Sunita and Milly the gallery also has many by Quintus. |
| | contradiction | | | 0.012 | |
| | neutral | X | X | 0.505 | |
| AS6 | entailment | | | 0.100 | sentence1<br>Sculptures inside the gallery include works by Terri and there are several large sculptures by Tomi displayed in the surrounding grounds.<br>sentence2<br>In addition to works by Terri and Tomi the gallery also has many by Fredrick. |
| | contradiction | | | 0.019 | |
| | neutral | X | X | 0.881 | |

Figure 4.3: Predictions and LIME explanations of each model on a *neutral* example from the MNLI dataset



| Model | Classes | Label | Pred | Score | Explanations |
|---|---|---|---|---|---|
| O | entailment | | | 0.052 | sentence1<br>We can save Fena Dim.<br>sentence2<br>Fema Dim is self-sufficient. |
| | contradiction | X | X | 0.636 | |
| | neutral | | | 0.312 | |
| AS1 | entailment | | | 0.170 | sentence1<br>We can save ENTITYNAME.<br>sentence2<br>ENTITYNAME is self-sufficient. |
| | contradiction | X | | 0.230 | |
| | neutral | | X | 0.600 | |
| AS2 | entailment | | | 0.048 | sentence1<br>We can save ENTITYNAME1.<br>sentence2<br>ENTITYNAME1 is self-sufficient. |
| | contradiction | X | | 0.374 | |
| | neutral | | X | 0.578 | |
| AS3 | entailment | | | 0.071 | sentence1<br>We can save ENTITYNAME2793.<br>sentence2<br>ENTITYNAME10253 is self-sufficient. |
| | contradiction | X | | 0.399 | |
| | neutral | | X | 0.530 | |
| AS4 | entailment | | | 0.096 | sentence1<br>We can save .<br>sentence2<br>is self-sufficient. |
| | contradiction | X | | 0.344 | |
| | neutral | | X | 0.560 | |
| AS5 | entailment | | | 0.032 | sentence1<br>We can save Nakeshia.<br>sentence2<br>Dannie is self-sufficient. |
| | contradiction | X | X | 0.588 | |
| | neutral | | | 0.379 | |
| AS6 | entailment | | | 0.029 | sentence1<br>We can save Ketan.<br>sentence2<br>Dannie is self-sufficient. |
| | contradiction | X | X | 0.823 | |
| | neutral | | | 0.148 | |

Figure 4.4: Predictions and LIME explanations of each model on a *Contradictory* example from the MNLI dataset

## 4.4 Discussion

Judging by the results of our experiments, we recommend practitioners to de-identify their sensitive textual data using random names, as they typically lead to the best results among the anonymisation strategies we tested. We also recommend to de-identify data before the training of NLP models. It follows that it is important to keep the de-identification process and naming schemes consistent throughout the entire pipeline that uses the data in order to mitigate potential performance losses of models. It may also be important to keep the number of names sufficiently high in order to avoid introducing bias in the training that may contribute to unfair discrimination against specific names, a well-known issue in machine learning models that handle person names [90].

## 4.5 Related Work

### 4.5.1 Impact of Anonymisation on Tabular Data

Various studies have been conducted to investigate the impact of anonymisation on tabular data. For instance, Slijepčević et al. [91] evaluated the impact of k-anonymisation [92] on downstream classification tasks using four tabular datasets such as the Adult dataset [93]. The authors observe decreases in performance as $k$ increases, but the amount of decrease was dependent on the chosen classifier and dataset.

### 4.5.2 Impact of Anonymisation on Textual Data

Relevant studies done on textual data largely focus on medical texts and on a very limited number of tasks and anonymisation strategies when compared to our work.

On the other hand, they typically anonymise a wide variety of protected health information (PHI) classes, while our work focuses on anonymisation of persons' names only. Berg et al. [86] studied the impact of four anonymisation strategies (pseudonymisation, replacement by PHI class, masking, and removal) on downstream NER tasks for the clinical domain. Similarly to our findings, they find that pseudonymisation yields the best results among the investigated strategies. On the other hand, removal of names resulted in the highest negative impact on the downstream tasks. Deleger et al. [87] investigated the impact of anonymisation on an information extraction task using a dataset of 3503 clinical notes. They anonymised 12 types of PHI such as patients' name, age, email address, etc., and used two anonymisation strategies (replacement by fake PHI, and masking). They found no significant loss in performance for this task. Similarly, Meystre et al. [81] found that the informativeness of medical notes only marginally decreased after anonymisation, using 18 types of PHI and 3 anonymisation strategies (replacement by fake PHI, replacement by PHI class, and replacement by generic *PHI* token). Using the same anonymisation strategies and ten types of PHI, Obeid et al. [94] investigated the impact of anonymisation on a mental status classification task. Comparing nine different machine learning models, they did not find any significant difference in performance between original and anonymised data.

## 4.6 Threats to Validity

As this study is limited in scope, there are certain threats to its validity. First, the de-identification is 1-dimensional as we only consider anonymising personal names while ignoring other kinds of personal data such as postal addresses, phone numbers, ID numbers, etc. On the other hand, this allowed us to investigate a wider variety of tasks where the datasets include only personal names and no other identifying information. Secondly, the scope of the study could have been widened by including more models, including non- Transformer-based ones. Finally, a bigger focus could have been given on the explainability aspect of the study. While we did find patterns in the decisions of the models using the LIT-tool and LIME, these patterns could have been coincidental in nature.

## 4.7 Summary

In this chapter, we presented an empirical study analysing the impact of text de-identification on downstream NLP tasks. We investigated the difference in performance of six distinct anonymisation strategies on nine NLP tasks ranging from simple classification tasks to hard NLU tasks. Furthermore, we compared two state-of-the-art architectures, those being the BERT and ERNIE architectures. Overall, we found that de-identifying data before training an NLP model does have a negative impact on its performance. However, this impact is relatively low. Furthermore, we determined that using pseudonymisation techniques that involve random names leads to higher performances across most investigated tasks. Specifically, replacing names by random names (AS5) had the least negative impact when using an ERNIE model. Similarly, replacing names by random names while preserving the link between identical names (AS6) worked best for simple BERT models.

In addition, we determined that it is advisable to also de-identify the data prior to training as we observed a large difference in performance between models trained on original data versus models trained on anonymised data.

We also observed a noticeable difference between the performances of BERT and ERNIE, warranting further investigation into the performance differences between a larger number of language models.

Finally, we conducted a limited qualitative analysis on our results using the Language Interpretability Tool. Our analysis suggested that using a generic token for the de-identification can lead to the misclassification of a given sample.

# PART II

## Multilingualism

In the second part, we study the challenges in multilingual systems. Specifically, we evaluate to what degree the presence of multiple languages affects the performance of such systems.

# 5 Comparing MultiLingual and Multiple MonoLingual Models for Intent Classification and Slot Filling

*In this chapter, we investigate the performance of conversational AI models, in particular in multilingual countries. Specifically, we investigate the strategies for training deep learning models of chatbots with multilingual data. We perform experiments for the specific tasks of Intent Classification and Slot Filling in financial domain chatbots and assess the performance of mBERT multilingual model vs multiple monolingual models.*

This chapter is based on the work published in the following research paper:

- Cedric Lothritz, Kevin Allix, Bertrand Lebichot, Lisa Veiber, Tegawendé F. Bissyandé, Jacques Klein. Comparing MultiLingual and Multiple MonoLingual Models for Intent Classification and Slot Filling, *International Conference on Applications of Natural Language to Information Systems*, 2021

## Contents

## 5.1 Overview

Chatbots usually operate in a single language depending on where they are deployed (e.g., a chatbot for a British bank will only handle requests written in English). While deploying a single monolingual chatbot is usually sufficient in countries where the entire population speaks one language, this strategy presents challenges in multilingual areas where people do not necessarily speak the same language at a high level. In multilingual countries, such as Switzerland, Luxembourg, India, South Africa, etc. with two or more national languages, companies and banks need to be able to communicate with their clients in the language of the latter's choosing in order to stay competitive. The same holds true for client support chatbots, which have to support multiple languages to stay viable in a multilingual environment. This requirement presents a challenge as companies have to decide on a strategy for implementing a multilingual chatbot system. Two such strategies are as follows: (S1) For $n$ languages, employ $n$ chatbots, each of which is trained to handle requests in a single language. (S2) For $n$ languages, employ one chatbot which is trained using data written in $n$ languages. There are some immediate advantages for training a chatbot using mixed-language data as one would have to train only a single chatbot and maintain only one database as opposed to multiple. However, it is unclear how the performance of a singular multilingual chatbot (S2) compares to a combination of multiple monolingual chatbots (S1).

In this chapter, we explore these two strategies for chatbots in a multilingual environment. Specifically, we investigate the performance of S1 and S2 on two tasks that represent fundamental blocks for chatbot systems: Intent Classification (IC), which is the task of identifying a user's intent based on a piece of text, and Slot Filling (SF), the task of identifying attributes that are relevant to a given intent. For this study, we use the Rasa chatbot framework, which uses the Dual Intent and Entity Transformer Classifier [95] for both the IC and SF tasks. Furthermore, we compare two techniques for text representation, namely bag-of-words (BOW) and multilingual BERT (mBERT) [3].

We aim to answer the following research questions:

1. RQ1: How does the number of languages affect the performance of Intent Classification and Slot Filling models?

2. RQ2: How does the distribution of data samples per language influence the performance of multilingual chatbots?

3. RQ3: How do S1 and S2 compare in terms of Intent Classification and Slot Filling?

For this study, we use a novel dataset for IC and SF in the financial domain, which we name the *Banking Client Support* (BCS) dataset. We also use the MultiATIS++ dataset published by Xu et al. [96].

The rest of this chapter is structured as follows: In Section 5.2, we explain the datasets we use, the chatbot framework, and give a detailed description for S1 and S2. In section 5.3, we present the results of our experiments, answer the research questions, and show the merits of multilingual chatbots. We discuss our findings in

Section 5.3.4 Section 5.4 shows various papers related to this study. In Section 5.5, we clarify some potential threats to the validity of our study, and we finally conclude our findings in Section 5.6.

## 5.2 Experimental Setup

In this section, we introduce the datasets that we use for this study, the chatbot framework and configuration, as well as the possible implementation strategy for multilingual chatbots.

### 5.2.1 Research Questions

1. RQ1: *How does the number of languages affect the performance of Intent Classification and Slot Filling models?* We use multilingual datasets for this study, and evaluate the difference in performance of models fine-tuned on subsets with varying numbers of languages.

2. RQ2: *How does the distribution of data samples per language influence the performance of multilingual chatbots?* We fine-tune models on bilingual datasets, and evaluate the difference in performance when varying the proportion of samples for each language.

3. RQ3: *How do S1 and S2 compare in terms of Intent Classification and Slot Filling?* We train pairs of models on bilingual datasets, one using the S1 strategy and one using the S2 strategy. We then compare the performance of each model.

### 5.2.2 Datasets

For this study, we use two multilingual datasets to evaluate the performance of multilingual chatbots. We created one dataset for client support bots in the banking domain as there are no public datasets available to the best of our knowledge. We also use a multilingual version of the well-known ATIS dataset to verify the results using a larger dataset.

#### 5.2.2.1 Banking Client Support Dataset

The first dataset (which we refer to as **B**anking **C**lient **S**upport dataset (BCS) throughout this chapter) is based on a toy dataset provided by Rasa[1]. The original dataset contains 337 samples divided into 15 intents. We removed three of the intents together with 93 samples as they seemed too vague (*inform*) or were not directly related to the banking domain (*help&human_handoff*), and added 763 samples and introduced 16 new intents, resulting in 1003 samples across 28 intents with each intent being distributed quite equally. The intents cover basic conversational phrases such as *greet* or *affirm* and requests specific to the banking domain such as *make_bank_transfer*, *block_card* or *search_atm*. Additionally, the set contains 253 entities, divided into 6 unique entity types such as *account_type* or *credit_card_type*. We then translated the dataset into three languages (German, French and Luxembourgish) with Google Translate and manually corrected translation errors, resulting in a total of four distinct, but parallel datasets[2]. Table 5.1

[1] `https://github.com/RasaHQ/financial-demo`

[2] Available at `https://github.com/Trustworthy-Software/BCS-dataset`

shows a small excerpt of each dataset. For this study, we use these four base datasets to construct mixed-language datasets containing equal numbers of samples from the base datasets, e.g., the English-French dataset consists of 50% English samples and 50% French data samples. There are 11 possible language combinations: six combinations with two languages, four with three languages, and one combination with all four languages, which gives us a total of 15 different datasets containing varying numbers of languages.

Table 5.1: Excerpt of the BCS dataset

| intent | English | French | German | Luxembourgish |
|---|---|---|---|---|
| greet | Good evening | Bonsoir | Guten Abend | Gudden Owend |
| close_account | Close my savings account, please. | Clôturez mon compte d'épargne, s'il vous plaît. | Bitte schließen Sie mein Sparkonto. | Maacht mäi Spuerkont zou, wann ech gelift. |
| order_card | I need a new credit card | J'ai besoin d'une nouvelle carte de crédit | Ich brauche eine neue Kreditkarte | Ech brauch eng nei Kreditkaart |
| apply_for_loan | I would like to. apply for a loan | Je souhaite demander un prêt. | Ich möchte einen . Kredit beantragen | Ech wéilt e Prêt ufroen. |

#### 5.2.2.2 MultiATIS++ Dataset:

The second dataset is based on the popular Airline Travel Information System (ATIS) dataset [53]. The original dataset contains a total of 5871 sentences divided into 26 intents. Furthermore, it contains 19 356 samples for slot filling, divided into 128 slot types. MultiATIS++ is a multilingual version of ATIS created by Upadhyay et al. [97] and Xu et al. [96]. For this study, we use the English, German and French versions of the MultiATIS++ dataset. Furthermore, we reduced the number of intents by removing intents with fewer than five samples, resulting in a total of 5860 sentences divided into 17 intents. Table 5.2 shows a small excerpt of the dataset. It is to note that the distribution of the intents is highly imbalanced with 73.6% of the samples having the intent *atis_flight*. There are four possible language combinations, resulting in a total of seven datasets.

Table 5.2: Excerpt of the MultiATIS++ dataset

| intent | English | French | German |
|---|---|---|---|
| airfare | All fares and flights from Philadelphia | Tous les tarifs et les vols de Philadelphia | Alle Tarife und Flüge von Philadelphia |
| flight | Show me flights from all airports to Love Field | Me montrer des vols de tous les aéroports à Love Field | Zeige mir die Flüge von allen Flughäfen nach Love Field |
| meal | What types of meals are available | Quels types de repas sont disponibles | Welche Arten von Mahlzeiten sind verfügbar |
| abbreviation | what does us stand for | Que signifie US | Was bedeutet US |

### 5.2.3 Chatbot Framework Used in this Study

Bocklisch et al. introduced the Rasa NLU and Rasa Core tools [95], with the objective of making a framework that is more accessible for creating conversational software. The modular design of a chatbot made with Rasa allows to swap out configuration files and training data. For this study, we created two different configurations: (C1) a bag-of-words (BOW, cf. Section 2.1.1) pipeline consisting of a WhitespaceTokenizer, RegexFeaturizer, LexicalSyntacticFeaturizer, and a CountVectorsFeaturizer. (C2) an mBERT pipeline which consists of the HFTransformersNLP model initializer using the cased multilingual *BERT Base* as its pre-trained model as well as its accompanying tokenizer and featurizer[3].

As our pre-trained language model, we use mBERT (cf. Section 2.1.3.3) as our datasets contain texts written in English, French, German, and Luxembourgish. However, as the number of Wikipedia articles varies greatly for every language of mBERT, there are significant disparities between the datasets used to train the different language components. Specifically, the English dataset is the largest with around 6 million articles, the German and French datasets have comparable sizes with 2.5 and 2.2 million articles respectively, and the Luxembourgish dataset is the smallest with only 59 000 articles. Table 5.3 shows the exact number of Wikipedia articles and words used for training mBERT for each language relevant to this study.

For this study, we use the cased mBERT model with 12 Transformer blocks, 768 hidden layers, attentions heads and 110 trainable parameters provided by Devlin et al.[4] [3].

Table 5.3: Training data size for mBERT

| | #Articles | #Words |
|---|---:|---:|
| English | 6 192 739 | 3 725 704 263 |
| German | 2 501 955 | 1 284 323 232 |
| French | 2 268 908 | 1 328 358 955 |
| Luxembourgish | 59 091 | 11 035 407 |

### 5.2.4 Implementation Strategies

Figure 5.1 shows the setup of multilingual and pseudo-multilingual chatbots trained on French and English data.

#### 5.2.4.1 S1: Pseudo-multilingual Chatbots

For each monolingual dataset, we train two chatbots: one using an mBERT model, and one without. By combining a language-selector (LS) and monolingual chatbots, we can create pseudo-multilingual chatbots (cf. Figure 5.1a). This allows us to directly compare the performance between monolingual chatbots and multilingual chatbots. For the language selector, we use either TextBlob[5] or langid[6]. TextBlob is able to identify Luxembourgish text as opposed to langid.

#### 5.2.4.2 S2: Multilingual Chatbots

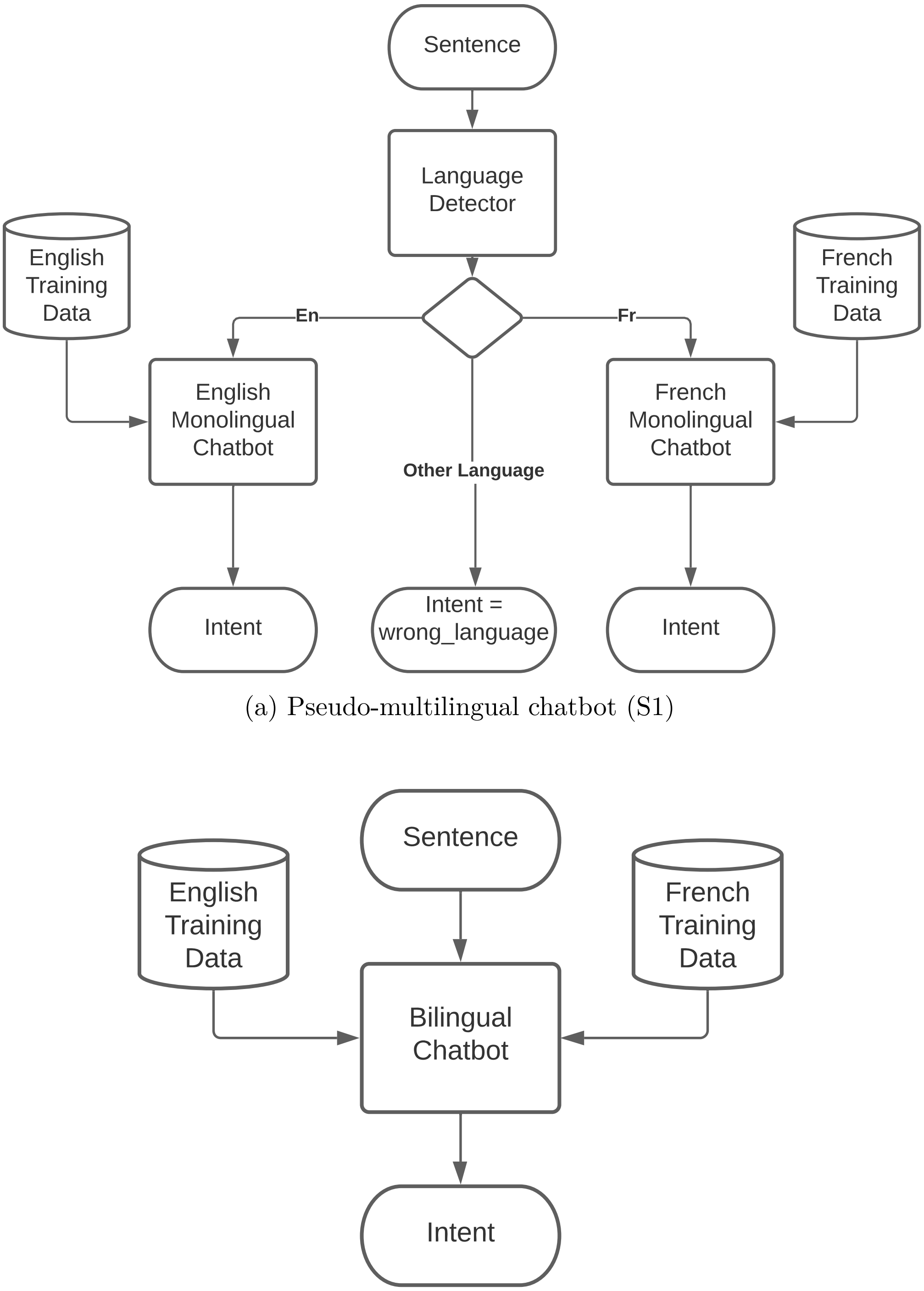
Based on the monolingual datasets, we construct mixed-language datasets. For every language combination, we extract a stratified subset from each monolingual dataset and combine them to create multilingual datasets. For each of these new datasets, we train two multilingual chatbots, one using a BOW model, and one using an mBERT model (cf. Figure 5.1b).

[3] Further information on Rasa models: `https://rasa.com/docs/rasa/components/`

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

[5] `https://github.com/sloria/TextBlob`

[6] `https://github.com/saffsd/langid.py`



(a) Pseudo-multilingual chatbot (S1)

(b) Multilingual chatbot (S2)

Figure 5.1: Setups for pseudo-bilingual and bilingual chatbots

## 5.3 Experimental Results

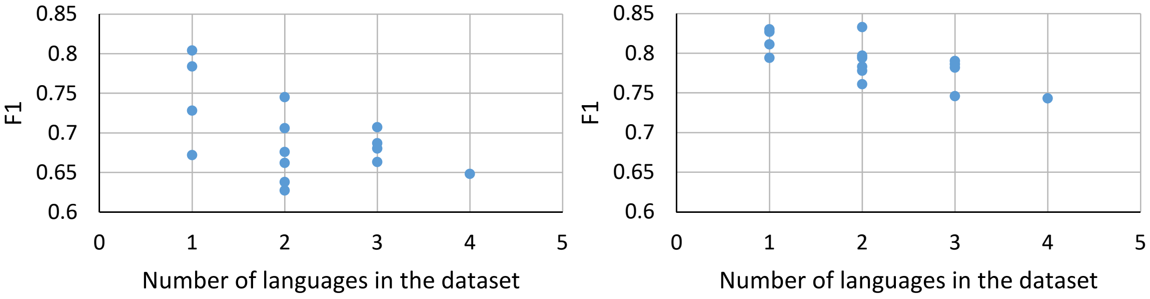
In this section, we will answer the three research questions that we formulated for this study (cf. Section 5.2.1).

### 5.3.1 RQ1: How does the number of languages affect the performance of Intent Classification and Slot Filling models?

In order to answer this question, we trained chatbot models in Rasa using the datasets that we constructed (cf. Section 5.2.2). We evaluated their performance for the IC and SF tasks using 5-fold cross-validation. Figure 5.2 shows the F1 scores for the IC task after training chatbots on the 15 various BCS datasets. Specifically, Figure 5.2a shows the performances of the chatbots using the mBERT model. While there are significant differences in the performance for every language, the plot seems to indicate a decrease in performance as the number of languages in the dataset increases.

Table 5.4: F1 scores for *Intent Classification* using the *BCS* datasets

| | 1 language | | | | 2 languages | | | | | | 3 languages | | | | 4 languages |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | En | Fr | De | Lb | En-Fr | En-De | En-Lb | Fr-De | Fr-Lb | De-Lb | En-Fr-De | En-Fr-Lb | En-De-Lb | Fr-De-Lb | En-Fr-De-Lb |
| mBERT | 0.804 | 0.784 | 0.728 | 0.672 | 0.745 | 0.706 | 0.676 | 0.627 | 0.662 | 0.638 | 0.707 | 0.687 | 0.663 | 0.680 | 0.648 |
| BOW | 0.830 | 0.827 | 0.811 | 0.794 | 0.833 | 0.797 | 0.794 | 0.761 | 0.778 | 0.783 | 0.773 | 0.773 | 0.807 | 0.785 | 0.743 |



(a) F1 scores with mBERT

(b) F1 scores with bag-of-word

Figure 5.2: Performances of chatbot models trained on the *BCS* datasets with $n$ languages for *Intent Classification* task. Each dot represents a different language combination.

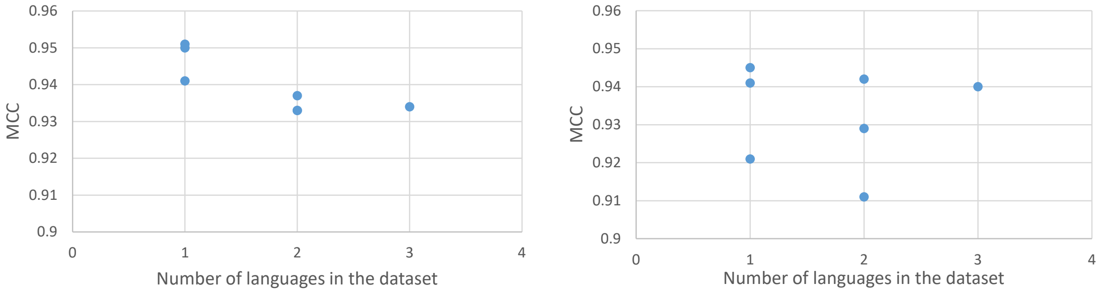
The performance for datasets containing English samples is generally higher than for the non-English datasets as is evidenced by Table 5.4. This can be explained by the amount of data used to pre-train mBERT. Conversely, the sets containing Luxembourgish samples usually lead to worse results overall, which can be due to the relatively small amount of data used to pre-train mBERT. This is consistent with the findings of Wu et al. [43]. Interestingly, there is a significant difference in performance between the French and German datasets, although mBERT was pre-trained on a similar number of articles for both languages.

A similar result can be observed for the MultiATIS++ set, albeit on a smaller scale. As the dataset is highly imbalanced, the F1-score can lead to an overly optimistic estimation of model performances [98]. As such, the Matthews Correlation Coefficient (MCC) is a more useful metric to evaluate the models' performances [99]. Indeed, while there is only a slight difference in performance regarding precision, recall and

F1, a larger difference can be observed for the MCC. Table 5.5 and Figure 5.3 show the results of Intent Classification for the MultiATIS++ datasets.

Table 5.5: F1 scores for *Intent Classification* using the *MultiATIS++* datasets

| | 1 language | | | 2 languages | | | 3 languages |
|---|---|---|---|---|---|---|---|
| | En | Fr | De | En-Fr | En-De | Fr-De | En-Fr-De |
| mBERT | 0.950 | 0.951 | 0.941 | 0.933 | 0.937 | 0.933 | 0.934 |
| BOW | 0.945 | 0.941 | 0.921 | 0.929 | 0.942 | 0.911 | 0.94 |



(a) F1 scores with mBERT

(b) F1 scores with bag-of-word

Figure 5.3: Performances of chatbot models trained on the *MultiATIS++* datasets with $n$ languages for *Intent Classification* task. Each dot represents a different language combination.

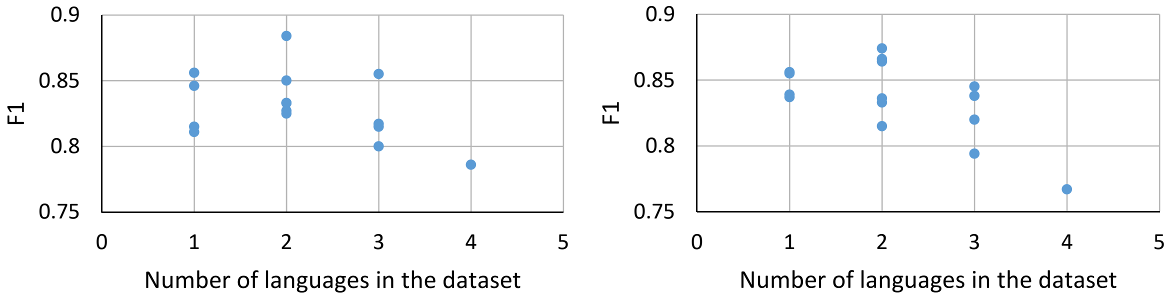
We observe that for mBERT models, the performance decreases as the number of languages increases. However, this trend is less obvious in the case of BOW models where the model trained on the En-Fr-De dataset performs similarly well as the one trained on the En-De dataset. Similarly to the BCS dataset, the models trained on data that include samples in English general yield higher performance than the models trained on purely non-English data.

Figure 5.2b shows the F1 scores for chatbot models trained without mBERT. Similarly to Figure 5.2a, it shows a decrease in performance as the number of languages in the training set increases, however, we can observe that the performances are better and the spread is smaller.

Figure 5.4 shows the results for slot filling using every language combination of the BCS dataset. Similarly to the IC-task, we observe that the performance of the models tends to decrease as the number of languages in the training set increases. This is true for both the mBERT models (see Figure 5.4a) and the BOW models (Figure 5.4b). We do see a similar result for models trained on MultiATIS++ datasets (Figure 5.5).

Table 5.6: F1 scores for *Slot Filling* with the *BCS* dataset

| | 1 language | | | | 2 languages | | | | | | 3 languages | | | | 4 languages |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | En | Fr | De | Lb | En-Fr | En-De | En-Lb | Fr-De | Fr-Lb | De-Lb | En-Fr-De | En-Fr-Lb | En-De-Lb | Fr-De-Lb | En-Fr-De-Lb |
| mBERT | 0.856 | 0.846 | 0.815 | 0.811 | 0.850 | 0.825 | 0.827 | 0.833 | 0.884 | 0.833 | 0.800 | 0.855 | 0.817 | 0.815 | 0.786 |
| BOW | 0.856 | 0.855 | 0.839 | 0.837 | 0.874 | 0.836 | 0.815 | 0.833 | 0.864 | 0.866 | 0.794 | 0.820 | 0.838 | 0.845 | 0.767 |

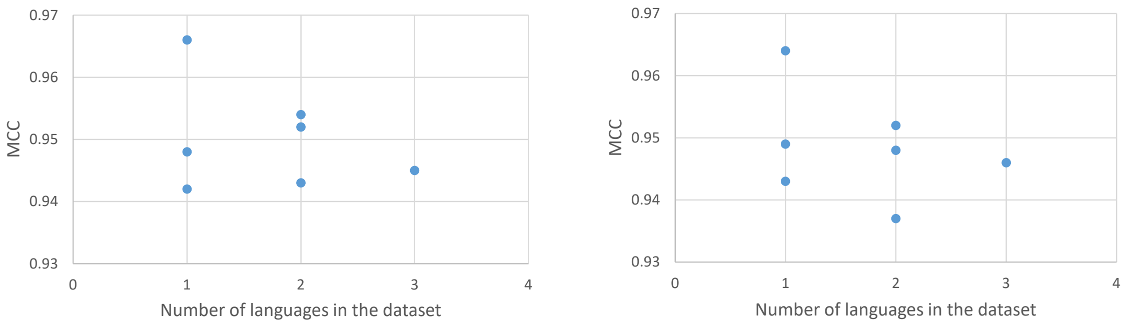


(a) F1 scores with mBERT (b) F1 scores with bag-of-word

Figure 5.4: Performances of chatbot models trained on the *BCS* datasets with $n$ languages for *Slot Filling* task. Each dot represents a different language combination.

Table 5.7: MCC scores for *Slot Filling* with the *MultiATIS++* dataset

| | 1 language | | | 2 languages | | | 3 languages |
|---|---|---|---|---|---|---|---|
| | En | Fr | De | En-Fr | En-De | Fr-De | En-Fr-De |
| mBERT | 0.966 | 0.948 | 0.942 | 0.952 | 0.954 | 0.943 | 0.945 |
| BOW | 0.964 | 0.943 | 0.949 | 0.948 | 0.952 | 0.937 | 0.946 |



(a) F1 scores with mBERT (b) F1 scores with bag-of-word

Figure 5.5: Performances of chatbot models trained on the *MultiATIS++* datasets with $n$ languages for *Slot Filling* task. Each dot represents a different language combination.

In contrast to the IC-task, for which the BOW models significantly outperformed the mBERT models, there is no clear favourite model for the Slot Filling task when using the BCS dataset. While the results show a smaller spread for the BOW models when compared to the mBERT models, the latter reach better performances for certain language combinations. Table 5.6 shows that BOW models consistently outperform mBERT when trained on monolingual datasets, but neither consistently outperforms the other when trained on mixed-language datasets. However, when trained on the MultiATIS++ datasets, we do see a clearer trend favouring the mBERT, even though the differences are relatively small.

**RQ1 Answer:** For both tasks, the performance of the models decreases as the number of languages in the training set increases.

### 5.3.2 RQ2: How does the distribution of data samples per language influence the performance of multilingual chatbots?

In order to answer this question, we create chatbots trained on bilingual datasets, vary the distribution of both languages in the sets, and evaluate their performance on various test sets. Specifically, we train 11 chatbot models on 11 mixed-language datasets where dataset 0 contains 0% samples from language A and 100% samples of language B, dataset 1 contains 10% samples of language A, 90% samples of language B, etc. These models are tested on three test sets: (1) a monolingual test set containing samples from language A, (2) a test set containing samples from language B, (3) a stratified test set containing an equal number of samples from both languages A and B.

#### 5.3.2.1 Intent Classification

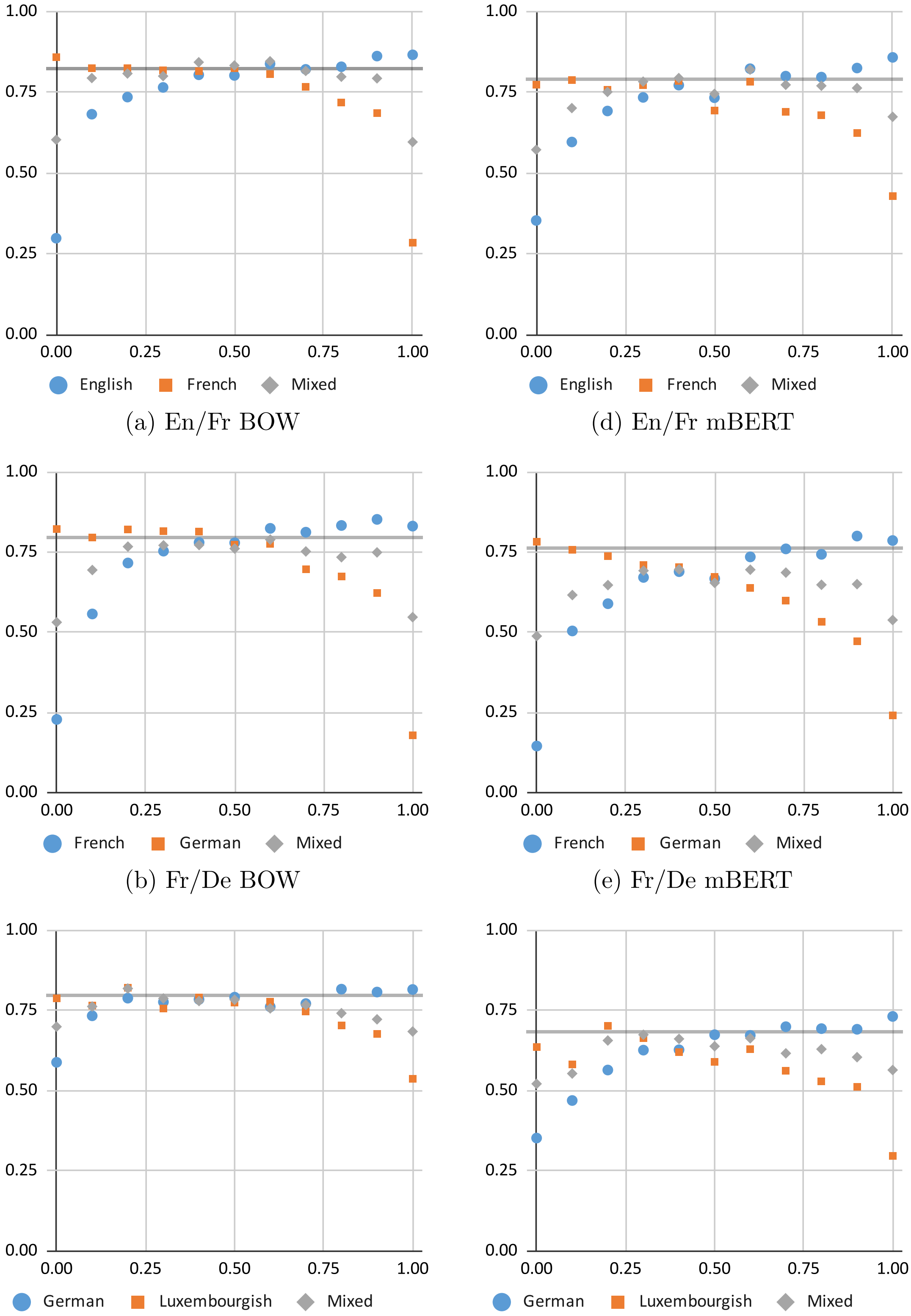
Figure 5.6 shows the performances of three language combinations in terms of F1 score. These combinations are: English/French (En/Fr), French/German (Fr/De) being two languages that are very dissimilar in terms of syntax and vocabulary, and German/Luxembourgish (De/Lb) being syntactically very similar. When varying the distribution of per-language data samples, we can make several observations: (1) when tested on a monolingual test set, we tend to observe very low performances if the training set does not contain the tested language at all, while we can see very high performances for the opposite case. This performance drop is less apparent for the De/Lb combinations (cf. Figure 5.6c and Figure 5.6f). Furthermore, the Fr/De combinations (cf. Figure 5.2b and Figure 5.6e) show the highest performance drop for these extreme cases. (2) When testing on the mixed-language test set, we can observe comparable performances for every training set, except for the models that were trained on monolingual training sets. (3) Models that are trained on sets containing 50% samples from each language tend to perform similarly for each test set. Figure 5.7 shows the results of the same experiment performed on the MultiATIS++ dataset. We observe that the performance remained stable except for the models trained on monolingual data.

#### 5.3.2.2 Slot Filling

Figure 5.8 shows the performances of the En/Fr, Fr/De, and De/Lb combinations for slot filling. We can make similar observations as we did for IC: we see very high and low performances for chatbots that were trained on monolingual datasets, with less noticeable drops for the German/Luxembourgish language combinations (cf. Figures 5.8c and 5.8f). When tested on the mixed test sets, most models perform similarly well except for the monolingual ones. It is to note that this performance drop is smaller for the SF task than it is for the IC task.

When performing the same experiment on the MultiATIS++ dataset (cf. Figure 5.9), the performance of the models fluctuated only slightly except for the models trained on monolingual data.

> **RQ2 Answer:** There is a noticeable drop in performance if a language is absent from the training set. A 50/50 split in the training set tends to lead to the highest performances on the mixed-language test sets.



(a) En/Fr BOW

(d) En/Fr mBERT

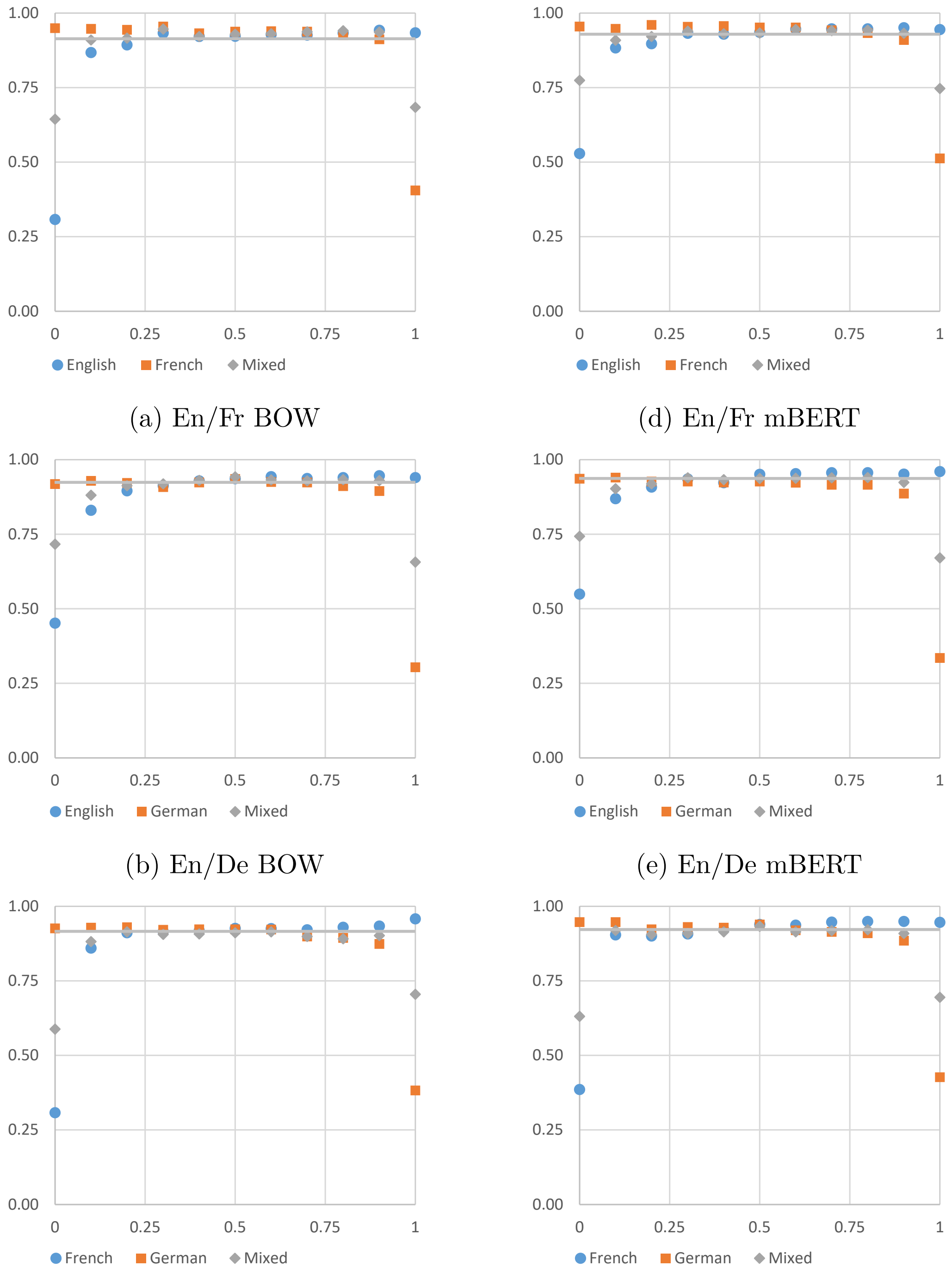
(b) Fr/De BOW

(e) Fr/De mBERT

(c) De/Lb BOW

(f) De/Lb mBERT

Figure 5.6: Evolution of the F1 score for bilingual chatbots for *Intent Classification* task using the *BCS* dataset when varying the distribution of data samples per language. The horizontal line represents the performance of the LS+monolingual chatbots models.



(a) En/Fr BOW

(d) En/Fr mBERT

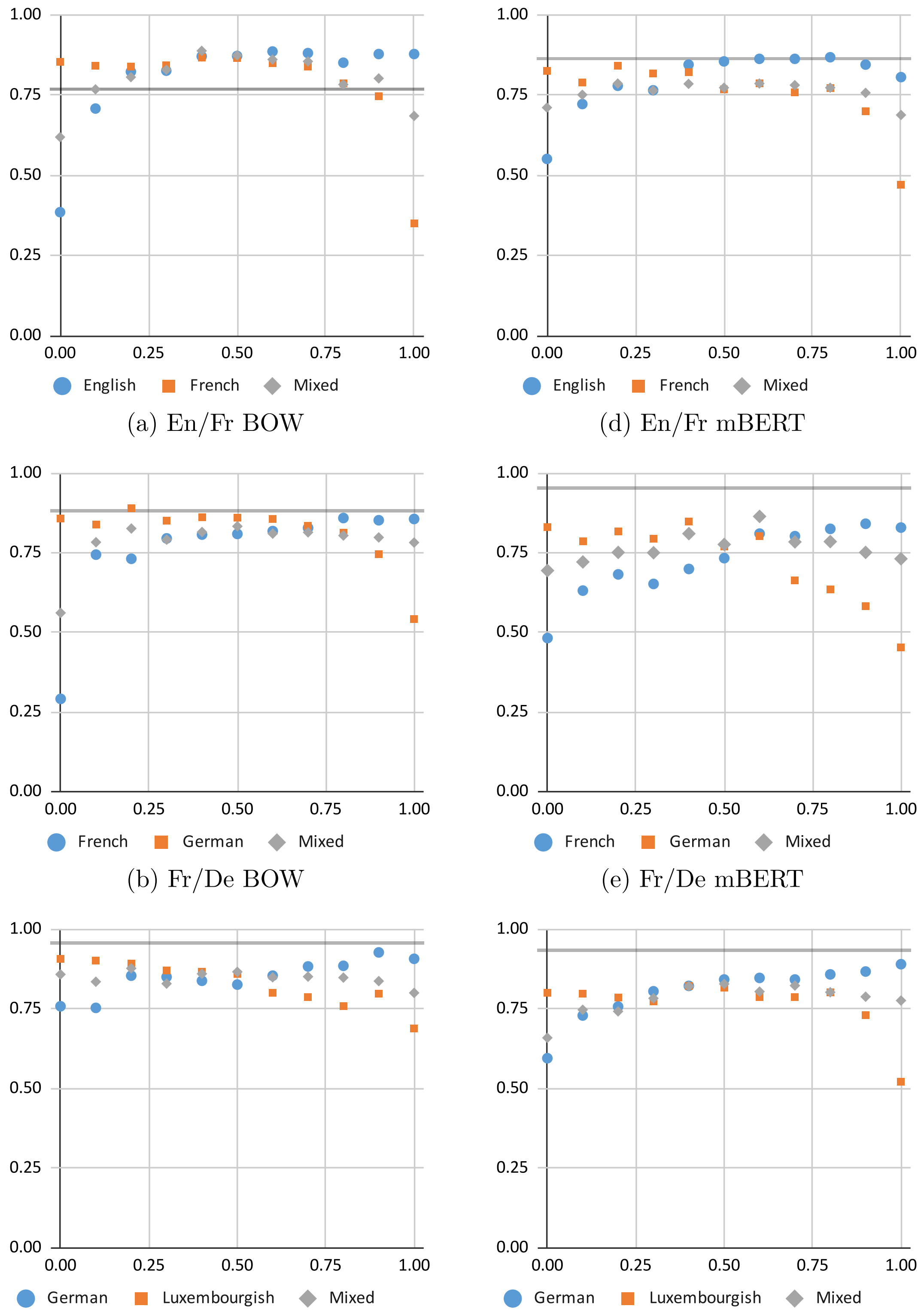
(b) En/De BOW

(e) En/De mBERT

(c) Fr/De BOW

(f) Fr/De mBERT

Figure 5.7: Evolution of the F1 score for bilingual chatbots for *Intent Classification* task using the *MultiATIS++* dataset when varying the distribution of data samples per language. The horizontal line represents the performance of the LS+monolingual chatbots models.



(a) En/Fr BOW

(d) En/Fr mBERT

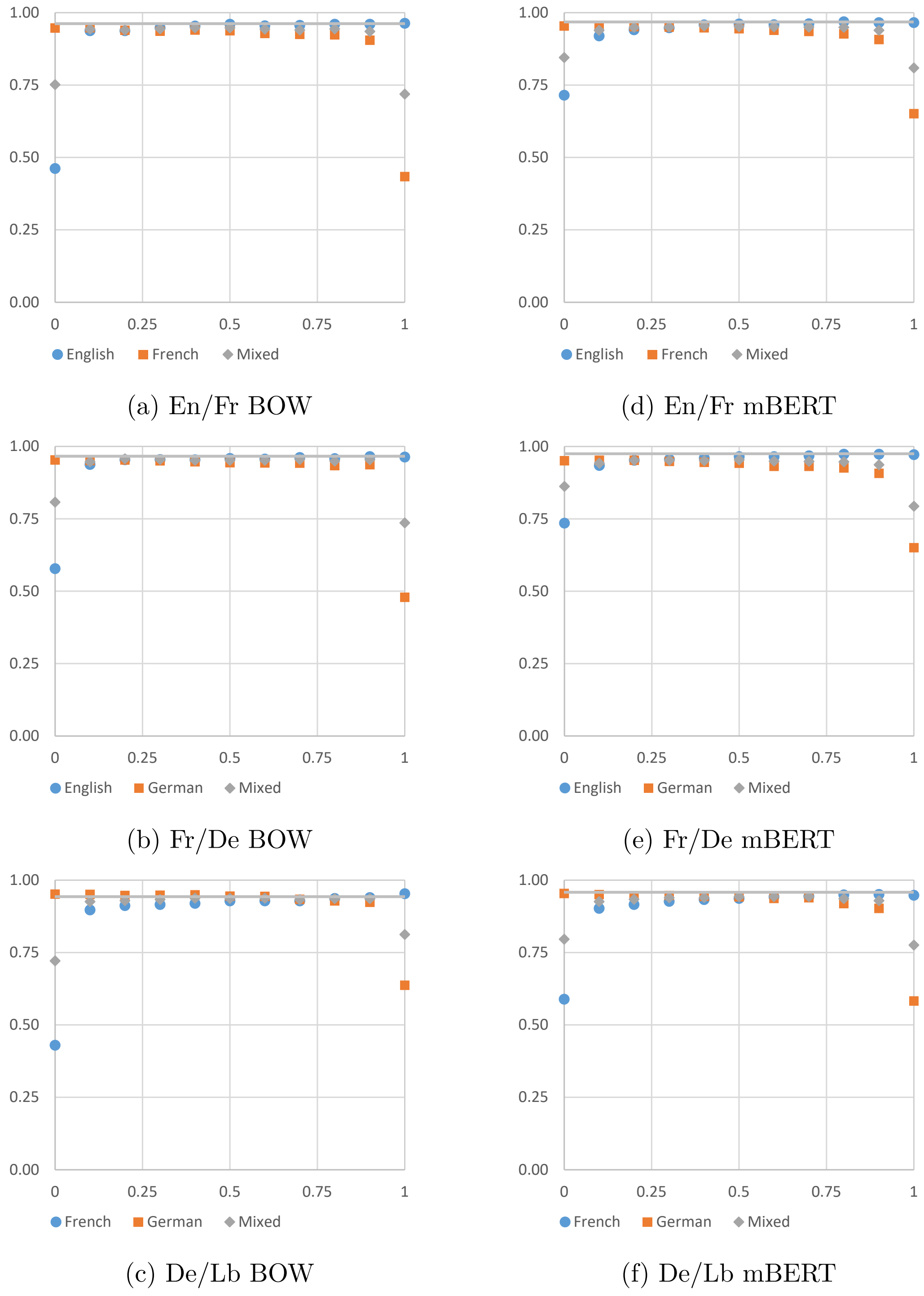
(b) Fr/De BOW

(e) Fr/De mBERT

(c) De/Lb BOW

(f) De/Lb mBERT

Figure 5.8: Evolution of the F1 score for bilingual chatbots for *Slot Filling* task using the *BCS* dataset when varying the distribution of data samples per language. The horizontal line represents the performance of the LS+monolingual chatbots models.



(a) En/Fr BOW

(d) En/Fr mBERT

(b) Fr/De BOW

(e) Fr/De mBERT

(c) De/Lb BOW

(f) De/Lb mBERT

Figure 5.9: Evolution of the F1 score for bilingual chatbots for *Slot Filling* task using the *MultiATIS++* dataset when varying the distribution of data samples per language. The horizontal line represents the performance of the LS+monolingual chatbots models.

### 5.3.3 RQ3: How do S1 and S2 compare in terms of Intent Classification and Slot Filling?

In order to answer this question, we reuse the bilingual chatbot models that were trained on the datasets which contain 50% data samples from each language (S2) and compare their performance to pseudo-bilingual chatbots (S1).

Table 5.8 compares F1 scores for pseudo-bilingual chatbot models and bilingual chatbot models for the IC task. Our results show that the combination of a language selector and two monolingual chatbots yields higher performances with regard to every performance measure used. It is to note that the English/French variant is an exception to the rule as the model with the S2 strategy significantly outperforms the S1 model. This trend can be observed for both the chatbot models with an mBERT and the ones with a BOW model. The performance differences between S1 and S2 models with mBERT are usually larger when compared to the performance differences between the models that do not use pre-trained models. Furthermore, the models based on BOW consistently outperform the models with mBERT by several percentage points.

Table 5.9 shows the results of the same task on the MultiATIS++ datasets. In contrast to the BCS sets, the results are in favour of the S2 strategy. When comparing the MCC scores, we observe that the performance of the bilingual models either exceeds or matches that of the combinations of LS+monolingual chatbots.

Table 5.8: Test results for bilingual chatbots (S2) vs monolingual chatbots with language selector (S1) on *Intent Classification* task on *BCS* set.

| | BOW | | | | | | mBERT | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Bilingual | | | LS + Monolingual | | | Bilingual | | | LS + Monolingual | | |
| | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 |
| En/Fr | 0.851 | 0.835 | **0.833** | 0.864 | 0.805 | 0.823 | 0.779 | 0.753 | 0.745 | 0.830 | 0.771 | **0.791** |
| En/De | 0.810 | 0.801 | 0.797 | 0.867 | 0.835 | **0.843** | 0.744 | 0.708 | 0.706 | 0.796 | 0.766 | **0.769** |
| En/Lb | 0.807 | 0.797 | 0.794 | 0.845 | 0.810 | **0.819** | 0.712 | 0.679 | 0.676 | 0.747 | 0.697 | **0.703** |
| Fr/De | 0.787 | 0.764 | 0.761 | 0.835 | 0.788 | **0.796** | 0.691 | 0.664 | 0.654 | 0.800 | 0.753 | **0.763** |
| Fr/Lb | 0.805 | 0.780 | 0.778 | 0.824 | 0.777 | **0.787** | 0.703 | 0.677 | 0.662 | 0.728 | 0.674 | **0.679** |
| De/Lb | 0.794 | 0.788 | 0.783 | 0.826 | 0.784 | **0.797** | 0.668 | 0.640 | 0.638 | 0.725 | 0.678 | **0.683** |

Table 5.9: Test results for bilingual chatbots(S2) vs monolingual chatbots with language selector(S1) on *Intent Classification* task on *MultiATIS++* set

| | BOW | | | | | | | | mBERT | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Bilingual | | | | LS + Monolingual | | | | Bilingual | | | | LS + Monolingual | | | |
| | Prec | Rec | F1 | MCC | Prec | Rec | F1 | MCC | Prec | Rec | F1 | MCC | Prec | Rec | F1 | MCC |
| En/Fr | 0.973 | 0.967 | 0.969 | **0.929** | 0.976 | 0.961 | 0.967 | 0.914 | 0.971 | 0.970 | 0.97 | **0.933** | 0.979 | 0.968 | 0.973 | 0.929 |
| En/De | 0.977 | 0.974 | 0.975 | **0.942** | 0.930 | 0.966 | 0.968 | 0.924 | 0.973 | 0.972 | 0.972 | **0.937** | 0.978 | 0.972 | 0.974 | **0.937** |
| Fr/De | 0.964 | 0.959 | 0.961 | 0.911 | 0.971 | 0.962 | 0.966 | **0.916** | 0.974 | 0.97 | 0.971 | **0.933** | 0.974 | 0.965 | 0.968 | 0.922 |

In order to determine if pseudo-bilingual (S1) significantly outperform bilingual (S2) models, we perform a Wilcoxon test for both strategies over every dataset used. We find that the differences in performance for mBERT models are indeed significant, but in the case for BOW models, only the difference in precision is clearly significant.

Tables 5.10 and 5.11 show the F1 scores with regard to the SF task. We generally see better results for the mBERT model. Similarly to the IC task, the combination of monolingual chatbots and a language selector almost consistently outperforms the chatbots trained on bilingual datasets by a large margin. This is true for both the BCS and the MultiATIS++ datasets.

Table 5.10: Test results for bilingual chatbots vs monolingual chatbots with language selector on *Slot Filling* task on *BCS* set

| | Bag-of-Words | | | | | | mBERT | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Bilingual | | | LS + Monolingual | | | Bilingual | | | LS + Monolingual | | |
| | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 |
| En/Fr | 0.877 | 0.882 | **0.874** | 0.86 | 0.712 | 0.768 | 0.841 | 0.728 | 0.773 | 0.955 | 0.810 | **0.863** |
| En/De | 0.804 | 0.894 | 0.836 | 0.928 | 0.911 | **0.919** | 0.919 | 0.802 | 0.839 | 0.969 | 0.920 | **0.943** |
| En/Lb | 0.765 | 0.898 | 0.815 | 0.904 | 0.814 | **0.848** | 0.835 | 0.724 | 0.760 | 0.977 | 0.900 | **0.931** |
| Fr/De | 0.847 | 0.841 | 0.833 | 0.890 | 0.877 | **0.882** | 0.864 | 0.721 | 0.776 | 0.993 | 0.917 | **0.953** |
| Fr/Lb | 0.817 | 0.928 | 0.864 | 0.917 | 0.927 | **0.921** | 0.898 | 0.797 | 0.825 | 1.000 | 0.883 | **0.935** |
| De/Lb | 0.856 | 0.884 | 0.866 | 0.975 | 0.950 | **0.956** | 0.890 | 0.796 | 0.828 | 0.950 | 0.926 | **0.934** |

Table 5.11: Test results for bilingual chatbots vs monolingual chatbots with language selector on *Slot Filling* task on *MultiATIS++* set

| | Bag-of-Words | | | | | | mBERT | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Bilingual | | | LS + Monolingual | | | Bilingual | | | LS + Monolingual | | |
| | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 |
| En/Fr | 0.948 | 0.949 | 0.948 | 0.962 | 0.965 | **0.962** | 0.953 | 0.952 | 0.952 | 0.969 | 0.970 | **0.968** |
| En/De | 0.953 | 0.955 | 0.952 | 0.966 | 0.967 | **0.966** | 0.954 | 0.955 | 0.954 | 0.975 | 0.976 | **0.975** |
| Fr/De | 0.937 | 0.940 | 0.937 | 0.946 | 0.945 | **0.943** | 0.946 | 0.944 | 0.943 | 0.959 | 0.959 | **0.958** |

We once again determine statistical significance of the obtained results through a Wilcoxon test. The resulting p-values show that the performance differences are significant except for recall and F1 score for the BOW models.

**RQ3 Answer:** In most cases, S1 performs better than S2, with IC on MultiATIS++ being a notable exception.

### 5.3.4 Discussion

When using a small dataset, the results of the conducted experiments are generally in favour of strategy S1 and by a significant margin. This is true for both the IC and the SF tasks. The results are less conclusive when training the chatbots on the larger MultiATIS++ dataset. For the IC task, neither strategy is consistently outperforming the other. On the other hand, strategy S1 is superior regardless of the dataset as it outperforms S2 for the BCS dataset as well as the MultiATIS++ dataset. The performances of the investigated models were significantly dependent on the task. While BOW-models generally performs better for the IC task, mBERT-models seems to be the favourable choice for the SF task, as strategy S1 with mBERT generally largely outperformed the BOW-models when compared directly.

## 5.4 Related Work

### 5.4.1 Multilingual Intent Classification and Slot Filling

Previous multilingual text classification systems are usually based on two different approaches: (1) machine translation systems that translate training data into the target language [100] or (2) parallel corpora that are used to learn embeddings jointly from multiple languages [101]. Such crosslingual embeddings prove useful for binary classification tasks such as Sentiment Classification [102, 103] and Churn Intent detection [104]. Abbet et al. [104] use multilingual embeddings for the task of churn intent detection in social media. They show that bilingual embeddings trained on an English and German dataset outperform monolingual embeddings for this binary IC task. Furthermore, they show that models trained on social media data can be applied to chatbot conversations as well. Schuster et al. [105] evaluate three methods for multilingual IC and SF, namely translating the training data into the target language, using pre-trained crosslingual embeddings, and using a novel pre-trained translation encoder to generate embeddings.

### 5.4.2 Multilingual Chatbots

Previous work often relied on machine-translation (MT) to create multilingual chatbots. Vanjani et al. [106] linked the Google Translate API to the English-speaking Rose chatbot[7], resulting in a bot that can converse in 103 different languages. A student talked the chatbot in German, and the resulting transcript was evaluated by 46 students in a Turing test. They concluded that the chatbot's utterances did not reach the same quality as a human's. In a similar study, Vanjani et al. [107] linked the Google Translate API to the Tutor Mike system[8] and evaluated transcripts given in German, Spanish and Korean for cogency and appropriateness. They found that while the replies given in German and Spanish were usually logical and natural, the quality of the Korean conversation was lower. Lin et al. [108] evaluated multilingual and crosslingual models for personalised dialogue systems and compared them to monolingual and MT-based models. They found that multilingual models performed better than MT-based models and similarly to monolingual models.

### 5.4.3 Multilingual Datasets

One major challenge for multilingual IC and SF is the lack of textual data in languages other than English. Schuster et al. created a dataset containing 57 000 utterances divided into three languages [105]: 43 000 utterances in English, 8600 in Spanish and 5000 in Thai. Their data is annotated for 12 intent types, and 11 slot types in total. They use their dataset to evaluate various crosslingual transfer methods for IC and SF. The ATIS dataset [53] is one of the most popular datasets for IC and SF. Originally available only in English, it was partially translated into Hindi and Turkish [109], creating MultiATIS. Xu et al. further extended MultiATIS to six more languages [96], resulting in MultiATIS++, consisting of nine versions of the original ATIS dataset. Datasets related to the banking domain are usually difficult to find as most of them are proprietary [110], making our BCS dataset one of the few public datasets related to that domain.

## 5.5 Threats to Validity

As this study is limited in scope, there are some potential shortcomings that threaten the validity of our observations. The first possible threat relates to the BCS dataset. As it is fairly small with only nearly 1000 samples, there is a possibility that our

[7] http://ec2-54-215-197-164.us-west-1.compute.amazonaws.com/speech.php

[8] http://bandore.pandorabots.com/pandora/talk?botid=ad1eeebfae345abc

models overfitted on the data. In addition, it was written by a small number of people, so the same writing style are repeated over most of the samples, adding on the possibility of overfitting. On the other hand, we repeated our experiments on the publicly available MultiATIS++ dataset, and found that our main findings remained largely consistent, confirming the results of our experiments on the BCS set. Furthermore, we could have included more architectures, as we consider only bag-of-words and mBERT for this study.

## 5.6 Summary

In this chapter, we presented a study on multilingual chatbots, specifically on the Intent Classification and Slot Filling tasks. We studied the effect of increasing the number of languages on the performance of the chatbot model. We also compared two implementation strategies and two embedding techniques. We noticed that training a chatbot on mixed-language data decreases the overall performance, and that the higher the number of languages in the dataset, the lower the performance in terms of F1 score. We concluded that, in the case of two languages, the combination of a language selector and two monolingual chatbots (S1) usually outperforms chatbots that are directly trained on bilingual datasets (S2). While the BOW models almost consistently outperform the mBERT models in the Intent Classification tasks, the mBERT models usually perform better in the Slot Filling tasks when using the S1 strategy.

# PART III

## Luxembourgish NLP

In the third part, we address the challenges related to the low-resource nature of the Luxembourgish language. In a first step, we focus on mitigating the lack of textual data by examining the usefulness of a novel data augmentation scheme for the creation of a Luxembourgish language model. In a second step, we examine the trade-offs of various pre-training schemes and use the gained knowledge to improve upon our Luxembourgish language model. We also mitigate the lack of annotated data by providing numerous Luxembourgish datasets for NLP tasks to the community.

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

*In this chapter, we present LuxemBERT, a BERT model for the Luxembourgish language that we create using a data augmentation approach based on partial translation. 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.*

This chapter is based on the work published in the following research paper:

- Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Lisa Veiber, Tegawendé F. Bissyandé, Jacques Klein, Andrey Boytsov, Clément Lefebvre, and Anne Goujon. LuxemBERT: Simple and Practical Data Augmentation in Language Model Pre-Training for Luxembourgish, *Proceedings of the Language Resources and Evaluation Conference*, 2022

## Contents

## 6.1 Overview

The increasing importance and popularity of pre-trained Language Models for NLP tasks over the last years is undeniable and they 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, as we established in Sections 2.1.3.2 and 2.1.3.3, BERT [3] and its variants such as RoBERTa [32], DistilBERT [111], and XLNet [31] are some of the most valuable contributions to the NLP community and 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. In Section 2.1.4.2, we brought up the fact that BERT models for English, German, French, and Spanish are trained on hundreds of millions of sentences. While this amount of data is readily available for such widely spoken languages, 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 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 [43]. To get enough data, Wu et al.[43] have 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 chapter, we introduce LuxemBERT, a BERT-like model for Luxembourgish. 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 corpus resemble more closely the limited corpus 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,

(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 chapter is structured as follows: In Section 6.2, we present our Luxem-

BERT model, the pre-training dataset we use, and the training hyperparameters. In Section 6.3, we define research questions for this study, present our baseline models, the datasets for the downstream tasks to evaluate LuxemBERT, and the fine-tuning parameters. In Section 6.4, we present the results of our experiments, address the research questions, and report the performance of LuxemBERT. Section 6.5 discusses the results we obtained. Section 6.6 discusses a selection of works related to this chapter. In Section 6.7, we present some potential threats to the validity of our study. Finally, we conclude our findings in Section 6.8.

## 6.2 LuxemBERT

Wu et al. [43] 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. Bernard et al. [112] 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.[1] Figure 6.1 shows the pre-training schema of our LuxemBERT model.

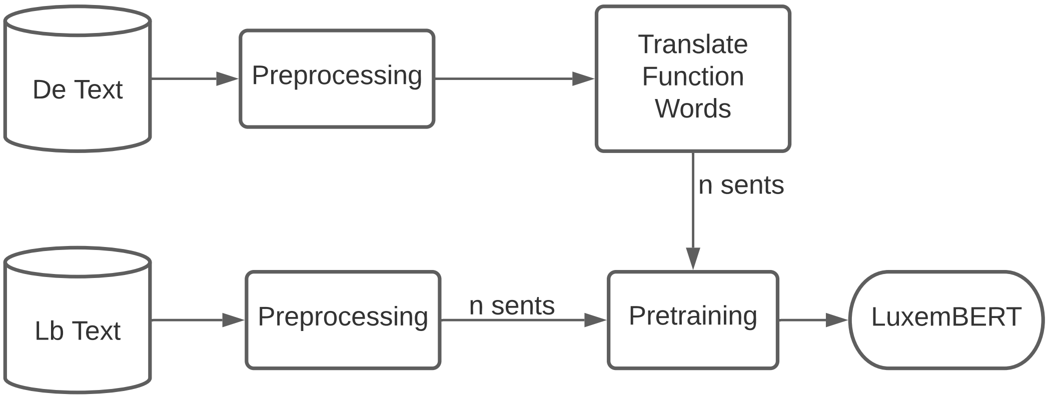


Figure 6.1: Data augmentation scheme for LuxemBERT (De: German / Lb: Luxembourgish)

For the creation of the pre-training corpus, we take advantage of the similarity between Luxembourgish and German. First, there is a sizeable overlap between the vocabularies between both languages. Indeed, we downloaded a list of 19 366 Luxembourgish-German word pairs[2], and determined that 3809 word pairs are identical, 3489 word pairs have a Levenshtein distance of 1 and 2333 pairs have a

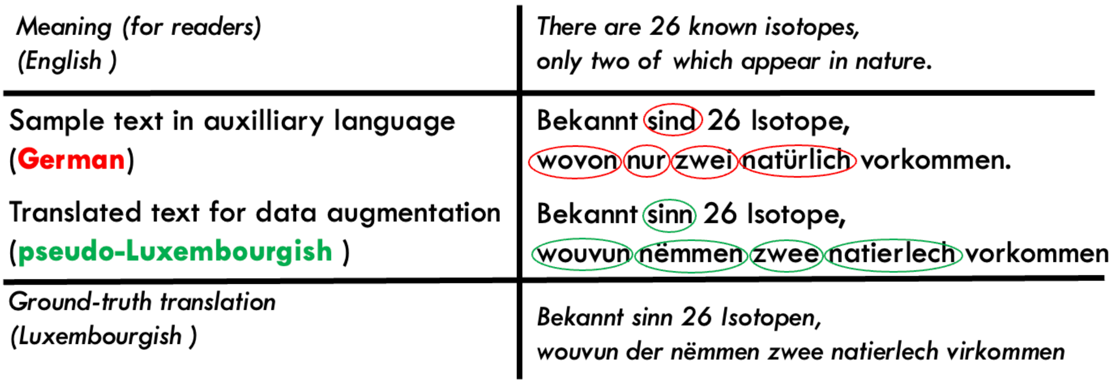
[1] The final (uncased) model can be found at `https://huggingface.co/lothritz/LuxemBERT`

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

distance of 2. Furthermore, both languages are closely related from a structural standpoint. The sentence syntax between both Luxembourgish and German is nearly identical with a few minor exceptions. Thanks to this syntactic similarity, 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 [113]. 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 et al. [114] 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 6.2 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.



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

Figure 6.2: Example pseudo-translation for LuxemBERT

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 6.3 shows the results of our experiment. While we find that the model pre-trained on 100% German data usually performs worst, the performances of 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.

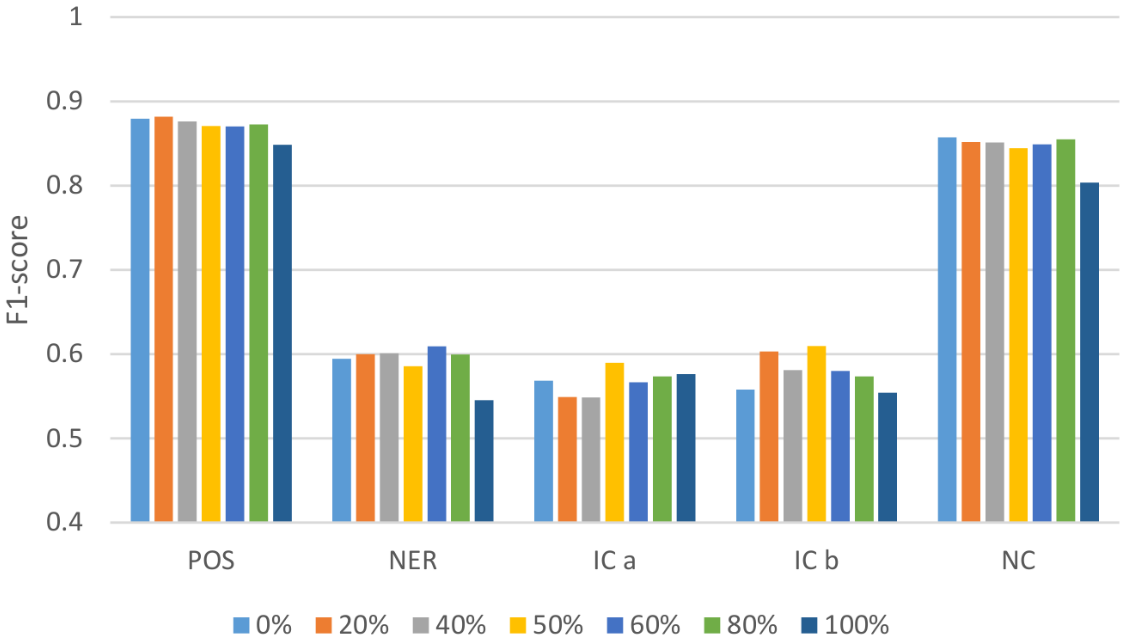


Figure 6.3: Results of experiments for determining best Lb/De ratio for LuxemBERT

### 6.2.1 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 6.1 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[3], 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 [45]. 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.[4] In addition, it contains pseudonymised chatlogs from the defunct Luxembourgish Chatroom Luxusbuerg and example sentences from the Luxembourgish Online Dictionary[5].

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

[4] www.rtl.lu

[5] www.lod.lu

We are aware of the OSCAR dataset [38] 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 pre-training.

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. [3] 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.

Table 6.1: Breakdown of pre-training corpus

| 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 |

#### 6.2.1.1 Pre-training Parameters

The *BERT Base* model created by Devlin et al. [3] 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 [115].

### 6.2.2 Cursory Evaluation of the MLM Task

In order to determine LuxemBERT's performance after pre-training, we perform a quick manual evaluation on the *Fill Mask* task. Given a sentence where a single word is masked out, we let the model predict a list of five suggestions to replace the masked word. We then manually evaluate whether the suggested words result in a sentence that is grammatically and semantically correct. Table 6.2 shows our selection of test sentences along with five words each proposed by the final, publicly available *uncased* LuxemBERT model as well as the *cased* model. We translated the sentences and suggestions into English for the reader's convenience. The suggestions are ordered by the model's confidence score. We highlighted in green the suggestions that are both grammatically and semantically correct.

Note that some suggestions are only valid in Luxembourgish, and that the English translations often introduce grammar or other mistakes. For example, the sentence "I go to retirement." is grammatically incorrect in English, but "Ech ginn an d'Pensioun." is a valid Luxembourgish sentence. The correct translation would be "I am retiring.".

Furthermore, note that we ignore factual errors. For instance, we accept the sentence "Paräis ass d'Haaptstad vu Spuenien." ("Paris is the Capital of Spain.") despite it being factually incorrect as it is correct from a grammatical and semantic standpoint. Finally, as mentioned in Section 1.2.3.3, many words in Luxembourgish have multiple valid spelling variations. This is also illustrated by the model's suggestion of "gin" and "ginn" for sentence 1, both of which mean "go" in this context, however, "gin" is the misspelled form of the word. In this work, we accept both spellings.

We observe that, overall, most of the uncased LuxemBERT's suggestions do indeed result in valid sentences. In terms of grammar, it appears to consistently suggest correct parts of speech. Furthermore, it seems capable of distinguishing genders of nouns. The words *en*, *e*, *um*, and *de* indicate that the noun that follows has to be male (examples 3, 4, 6, 10), while the words *d'*, and *eng* precede a female noun (examples 2, 5). Uncased LuxemBERT also almost consistently applies the *Äifler Regel* correctly, which is a phonological rule where words lose their final *n* if they are followed by certain words [116]. We can observe this rule being applied in the sentences 3 and 4, where the Äifler Regel dictates that the word following "en" has to begin with one of the letters in $\{a, e, i, o, u, y, d, h, n, t, z\}$, while the word following "e" cannot begin with any of these letters. The model applied the rule correctly in all cases except for *Kaffi* in sentence 3 as the correct use would be "e Kaffi" rather than "en Kaffi".

With regard to semantics, the uncased LuxemBERT usually suggests words that make sense in the context of the sentence. Once notable exception is that the model oftentimes suggests drinks rather than food (examples 3 and 4). Finally, it appears that it learned some world knowledge during the pre-training phase indicated by its most confident suggestions for examples 8, 9, and 10.

On the other hand, the *cased* version of LuxemBERT seemed to struggle with half of the sentences, in particular sentences 2 through 6. It almost consistently suggested single characters rather than words. Interestingly, it seemed to "correctly apply" the Äifler Regel as the letters suggested in sentence 2 are indeed ones that must follow the word "en".

We note that in the cases where the cased model did predict entire words, they were usually correct from both a grammatical and a semantic standpoint.

Table 6.2: Selection of sentences to test the pre-training task of LuxemBERT. Suggestions to replace the [MASK] tokens are ordered by confidence score. We highlighted in green whether a suggestion is valid.

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | gin | ginn | muss | war | fueren |
| | En | I [MASK] to school. | go | go | have to (go to) | was | drive |
| 2 | Lb | Ech ginn an d'[MASK]. | Schoul | Pensioun | Vakanz | Politik | Kierch |
| | En | I go to the [MASK]. | school | retirement | holiday | politics | church |
| 3 | Lb | Ech iessen en [MASK]. | net | och | Kaffi | Téi | Déier |
| | En | I eat a/it [MASK]. | not | also | coffee | tea | animal |
| 4 | Lb | Ech iessen e [MASK]. | Kaffi | Kuch | Patt | Fleesch | Béier |
| | En | I eat a [MASK]. | coffee | cake | drink | meat | beer |
| 5 | Lb | Ech iessen eng [MASK]. | Glace | Pizza | Zopp | Kou | Schmier |
| | En | I eat a [MASK]. | ice cream | pizza | soup | cow | sandwich |
| 6 | Lb | Den Hond läit um [MASK]. | Buedem | Kapp | Réck | Bauch | Hals |
| | En | The dog lies on the/its [MASK]. | floor | head | back | belly | neck |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Diddeleng | Esch | Beetebuerg | Schëffleng | Péiteng |
| | En | The (motorway) A4 connects Esch and [MASK]. | Dudelange | Esch | Bettembourg | Schifflange | Pétange |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Frankräich | Paräis | Spuenien | Bordeaux | Versailles |
| | En | Paris is the Capital of [MASK]. | France | Paris | Spain | Bordeaux | Versailles |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Paräis | Et | Versailles | Arrondissement | Si |
| | En | [MASK] is the Capital of France. | Paris | it | Versailles | arrondissement | it |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Buergermeeschter | Premier | Premierminister | Spëtzekandidat | Fraktiounschef |
| | En | Xavier Bëttel is the [MASK] of Luxembourg. | mayor | prime minister | prime minister | frontrunner | faction leader |

(a) Suggestions for LuxemBERT *uncased*

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | ginn | gin | muss | war | fueren |
| | En | I [MASK] to school. | go | go | have to (go to) | was | drive |
| 2 | Lb | Ech ginn an d'[MASK]. | f | k | b | r | s |
| | En | I go to the [MASK]. | f | k | b | r | s |
| 3 | Lb | Ech iessen en [MASK]. | net | t | a | z | h |
| | En | I eat a/it [MASK]. | not | t | a | z | h |
| 4 | Lb | Ech iessen e [MASK]. | k | g | b | s | v |
| | En | I eat a [MASK]. | k | g | b | s | v |
| 5 | Lb | Ech iessen eng [MASK]. | s | k | t | p | b |
| | En | I eat a [MASK]. | s | k | t | p | b |
| 6 | Lb | Den Hond läit um [MASK]. | Kapp | Kënn | 1 | p | a |
| | En | The dog lies on the/its [MASK]. | head | chin | 1 | p | a |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Lëtzebuerg | Ell | Éinen | Feelen | Käl |
| | En | The (motorway) A4 connects Esch and [MASK]. | Luxembourg | Ell | Ehnen | Feulen | Kayl |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Lëtzebuerg | Wien | Berlin | Japan | St |
| | En | Paris is the Capital of [MASK]. | Luxembourg | Vienna | Berlin | Japan | St |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Et | Si | Lëtzebuerg | Dat | Se |
| | En | [MASK] is the Capital of France. | it | it | Luxembourg | that | it |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Premier | Mann | Chef | Minister | Bierger |
| | En | Xavier Bëttel is the [MASK] of Luxembourg. | prime minister | man | boss | minister | citizen |

(b) Suggestions for LuxemBERT *cased*

## 6.3 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.

### 6.3.1 Research Questions

We investigate the following research questions:

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

## 6.3.2 Baseline Models

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

### 6.3.2.1 mBERT

As LuxemBERT is the first Transformer-based model for the Luxembourgish language, we use the mBERT (cf. Section 2.1.3.3)[6] as a baseline to evaluate the performance of the LuxemBERT models on the selected downstream tasks. 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 time of training.

### 6.3.2.2 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.

### 6.3.2.3 Lb/De BERT: Bilingual BERT

Following the approach by Wu et al. [43], 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.

## 6.3.3 Downstream Tasks

We consider five downstream 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 (cf. Section 2.2). As suitable datasets are scarce, we create a number of Luxembourgish ones ourselves. Table 6.3 shows an overview of each dataset used for fine-tuning. As most of these datasets are based on articles from RTL, we cannot publish them, but we make them available to researchers on request.

Table 6.3: Breakdown of datasets used for fine-tuning LuxemBERT on downstream tasks

| Task | train | dev | test | #labels | max | min | mean | median |
|---|---|---|---|---|---|---|---|---|
| POS | 4291 | 459 | 750 | 15 | 16452 | 7 | 4864 | 3915 |
| NER | 4291 | 459 | 750 | 5 | 2272 | 95 | 1214 | 1239 |
| IC a | 698 | 149 | 159 | 28 | 60 | 23 | 36 | 35.5 |
| IC b | 606 | 130 | 137 | 23 | 60 | 28 | 38 | 37 |
| NC | 7057 | 1034 | 1961 | 8 | 2866 | 106 | 1257 | 1120 |
| WNLI | 568 | 63 | 136 | 2 | 409 | 358 | 383.5 | 383.5 |

### 6.3.3.1 Part-of-Speech Tagging

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

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

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 ground-truth dataset was done using a Luxembourgish spaCy model[7] 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.

#### 6.3.3.2 Named Entity Recognition

For the NER task, 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.

#### 6.3.3.3 Intent Classification

For the IC task, we reuse the Banking Client Support dataset (cf. Section 5.2.2.1). 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.

#### 6.3.3.4 News Classification

To build the NC dataset, 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.

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

#### 6.3.3.5 Winograd Natural Language Inference

For this dataset, we modified the WNLI dataset that was originally created by Levesque et al. [61]. We translated the dataset to Luxembourgish[8]. 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.

### 6.3.4 Fine-tuning Parameters

Regarding fine-tuning parameters, Devlin et al. [3] report that the best performances for downstream NLP tasks are observed for a batch size in $\{16, 32\}$, a learning rate in {2e-5,3e-5,5e-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 6.4.

Table 6.4: Results of grid search for parameters

| Task | batch size | learning rate | #epochs |
|---|---|---|---|
| POS | 16 | 5e-5 | 3 |
| NER | 16 | 5e-5 | 3 |
| IC a | 16 | 5e-5 | 5 |
| IC b | 16 | 5e-5 | 5 |
| NC | 16 | 2e-5 | 2 |
| WNLI | 16 | 5e-5 | 5 |

## 6.4 Experimental Results

In this section, we present and analyse the results of our experiments and answer the research questions we asked in Section 6.3.1. As mentioned in Section 6.2.1, 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 6.5 and 6.6 show 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 [117]. This test is not very powerful [118] but allows to compare all pairs of models directly and has an easy-to-interpret visualization. It 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.

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

Table 6.5: Comparison of results for *uncased* models on downstream tasks

| 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 6.6: Comparison of results for *cased* 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 |

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** [117] compares the difference of performance for a pair of approaches across datasets. It is more powerful than F/N tests as it only considers two alternatives.

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

First, we do the same cursory evaluation of the *Fill Mask* task that we performed on the LuxemBERT models in Section 6.2.2, but we evaluate the uncased and cased mBERT models. Table 6.7 shows the results of this experiment.

We observe that both mBERT models struggle to make proper suggestions for the sentences 1 to 6, typically failing to even suggest proper Luxembourgish words and instead resorting to single characters, suffixes (marked by ##), or non-Luxembourgish words. On the other hand, they usually manage to produce meaningful sentences for sentences 7 through 10. It is to note that these sentences can be regarded as statements of facts which are commonly found in texts such as encyclopedias, and by extension, Wikipedia articles. For that reason, it is unsurprising that a language model trained on this kind of data would handle those kinds of sentences well. This first set of experiments seems to indicate that the LuxemBERT models handle the Luxembourgish language better than the mBERT models.

Figure 6.4 shows a comparison of both mBERT and LuxemBERT models. With regard 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

Table 6.7: Performance of the uncased and cased mBERT BERT models. Suggestions to replace the [MASK] tokens are ordered by confidence score. We highlighted in green whether a suggestion is valid. ## marks a suffix for the previous word.

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | ##ill | ##uel | ##ed | ##eh | ##ap |
| | En | I [MASK] to school. | ##ill | ##uel | ##ed | ##eh | ##ap |
| 2 | Lb | Ech ginn an d'[MASK]. | Stad | u | z | 2 | nr |
| | En | I go to the [MASK]. | town | u | z | 2 | nr |
| 3 | Lb | Ech iessen en [MASK]. | nederland | Europa | zee | belgie | Vlaanderen |
| | En | I eat a/it [MASK]. | Netherlands | Europe | sea | Belgium | Flanders |
| 4 | Lb | Ech iessen e [MASK]. | ##w | s | al | ! | man |
| | En | I eat a [MASK].##w | s | old | ! | man | |
| 5 | Lb | Ech iessen eng [MASK]. | ##r | ##h | ##s | ##l | ##g |
| | En | I eat a [MASK]. | ##r | ##h | ##s | ##l | ##g |
| 6 | Lb | Den Hond läit um [MASK]. | km$^2$ | m | km | hektar | Island |
| | En | The dog lies on the/its [MASK]. | km$^2$ | m | km | hektar | Iceland |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Lëtzebuerg | Köln | Luxemburg | Aachen | Koblenz |
| | En | The (motorway) A4 connects Esch and [MASK]. | Luxembourg | Cologne | Luxembourg | Aachen | Koblenz |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Lëtzebuerg | frans | paris | Frankreich | france |
| | En | Paris is the Capital of [MASK]. | Luxembourg | French | Paris | France | France |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Metz | Nancy | Toulouse | Troyes | Poitiers |
| | En | [MASK] is the Capital of France. | Metz | Nancy | Toulouse | Troyes | Poitiers |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Politiker | President | Gouverneur | maire | Premierminister |
| | En | Xavier Bëttel is the [MASK] of Luxembourg. | politician | president | governor | mayor | prime minister |

(a) Suggestions for mBERT *uncased*

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | , | ##ch | ##lo | war | ... |
| | En | I [MASK] to school. | , | ##ch | ##lo | was | ... |
| 2 | Lb | Ech ginn an d'[MASK]. | St | d | D | Dr | H |
| | En | I go to the [MASK]. | St | d | D | Dr | H |
| 3 | Lb | Ech iessen en [MASK]. | op | is | de | en | in |
| | En | I eat a/it [MASK]. | entirely | is | the | a/it | in |
| 4 | Lb | Ech iessen e [MASK]. | ##ch | ##h | dr | ##hn | St |
| | En | I eat a [MASK]. | ##ch | ##h | dr | ##hn | St |
| 5 | Lb | Ech iessen eng [MASK]. | ##e | ##en | ##er | . | ##n |
| | En | I eat a [MASK]. | ##e | ##en | ##er | . | ##n |
| 6 | Lb | Den Hond läit um [MASK]. | Land | K | Vol | s | k |
| | En | The dog lies on the/its [MASK]. | countryside | K | flight | s | k |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Esch | Thorn | Antwerpen | Bus | Luxemburg |
| | En | The (motorway) A4 connects Esch and [MASK]. | Esch | Thorn | Antwerpen | Bus | Luxembourg |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Lëtzebuerg | Par | Paris | Kantonen | Republik |
| | En | Paris is the Capital of [MASK]. | Luxembourg | Par | Paris | cantons | republic |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Lëtzebuerg | Esch | Et | Arrondissement | Gare |
| | En | [MASK] is the Capital of France. | Lëtzebuerg | Esch | it | arrondissement | train station |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Fränk | Premierminister | Xavier | Politiker | Jong |
| | En | Xavier Bëttel is the [MASK] of Luxembourg. | Frank | prime minister | Xavier | politician | boy |

(b) Suggestions for mBERT *cased*

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.

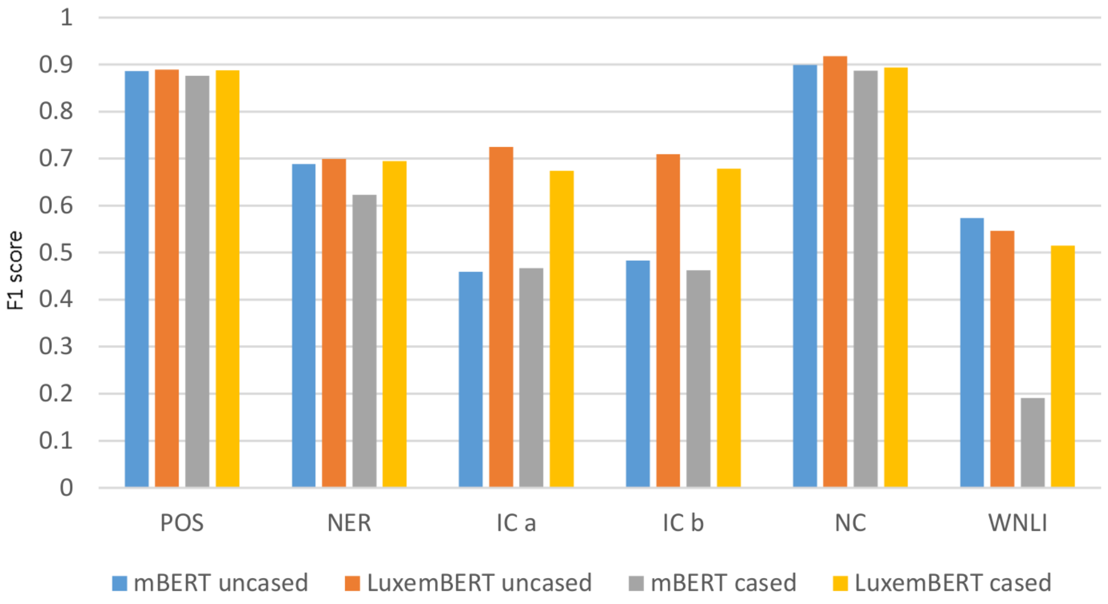


Figure 6.4: mBERT vs LuxemBERT

### 6.4.2 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. As a first comparison, we evaluate the performance of the cased and uncased models on the MLM task by using the same 10 example sentences from Section 6.2.2. Tables 6.7 and 6.8 show the suggestions of each model.

We observe that the De/Lb models often fail to produce meaningful answers, resorting to single character suggestions instead. This is in particular true for sentences 3 to 5. We do notice that, similarly to the cased LuxemBERT model, both De/Lb models "apply" the Äifler Regel correctly despite producing mostly single character responses. For this reason, we conclude that this behaviour was learned during the pre-training and is not coincidental.

The performance of Lb BERT models is comparable to that of the LuxemBERT models, with the suggestions of *uncased* Lb BERT being almost always correct from both a grammatical and semantic standpoint, and thus matching the *uncased* LuxemBERT model in that regard. However, the *cased* Lb BERT model performs significantly better than the *cased* LuxemBERT model, managing to produce meaningful suggestions for each sentence except for example 5.

Overall, it appears that both the uncased Lb BERT and uncased LuxemBERT models perform best in this experiment.

Table 6.8: Performance of the uncased and cased Lb BERT models. Suggestions to replace the [MASK] tokens are ordered by confidence score. We highlighted in green whether a suggestion is valid.

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | ginn | gin | muss | fueren | kommen |
|  | En | I [MASK] to school. | go | go | have to (go to) | drive | come |
| 2 | Lb | Ech ginn an d'[MASK]. | Vakanz | Pensioun | Schoul | Ausland | Bett |
|  | En | I go to the [MASK]. | holiday | retirement | school | foreign countries | bed |
| 3 | Lb | Ech iessen e [MASK]. | net | och | Téi | nie | do |
|  | En | I eat a/it [MASK]. | not | also | tea | never | there |
| 4 | Lb | Ech iessen e [MASK]. | Kaffi | Fësch | Kuch | Croissant | Béier |
|  | En | I eat a [MASK]. | coffee | fish | cake | croissant | beer |
| 5 | Lb | Ech iessen eng [MASK]. | Ham | Pizza | Glace | Zalot | Zopp |
|  | En | I eat a [MASK]. | ham | pizza | ice cream | salad | soup |
| 6 | Lb | Den Hond läit um [MASK]. | Buedem | Kapp | Bauch | Réck | Hals |
|  | En | The dog lies on the/its [MASK]. | floor | head | belly | back | neck |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Schëffleng | Diddeléng | Esch | Beetebuerg | Péiteng |
|  | En | The (motorway) A4 connects Esch and [MASK]. | Schifflange | Dudelange | Esch | Bettembourg | Pétange |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Paräis | Frankräich | Lëtzebuerg | Bréissel | London |
|  | En | Paris is the Capital of [MASK]. | Paris | France | Luxembourg | Brussels | London |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Et | Lëtzebuerg | Si | Schengen | Paräis |
|  | En | [MASK] is the Capital of France. | it | Luxembourg | it | Schengen | Paris |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Premier | President | Premierminister | Buergermeeschter | Staatschef |
|  | En | Xavier Bëttel is the [MASK] of Luxembourg. | prime minister | president | prime minister | mayor | head of state |

(a) Suggestions for Lb BERT *uncased*

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | ginn | gin | fueren | war | muss |
|  | En | I [MASK] to school. | go | go | drive | was | have to (go to) |
| 2 | Lb | Ech ginn an d'[MASK]. | Vakanz | Politik | Schoul | Kierch | Mass |
|  | En | I go to the [MASK]. | holidays | politics | school | church | church mass |
| 3 | Lb | Ech iessen en [MASK]. | net | och | Déier | Haus | 2 |
|  | En | I eat a/it [MASK]. | not | also | animal | house | 2 |
| 4 | Lb | Ech iessen e [MASK]. | bëssen | wéineg | net | gutt | Stéck |
|  | En | I eat a [MASK]. | little | little | also | good | piece |
| 5 | Lb | Ech iessen eng [MASK]. | Hand | 2 | Mask | 1 | 3 |
|  | En | I eat a [MASK]. | hand | 2 | mask | 1 | 3 |
| 6 | Lb | Den Hond läit um [MASK]. | Réck | Kapp | Buedem | Been | Waasser |
|  | En | The dog lies on the/its [MASK]. | back | head | floor | leg | water |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Esch | Lëtzebuerg | Schengen | Gare | Feelen |
|  | En | The (motorway) A4 connects Esch and [MASK]. | Esch | Luxembourg | Schengen | train station | Feulen |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Lëtzebuerg | Japan | Wien | St | Schengen |
|  | En | Paris is the Capital of [MASK]. | Luxembourg | Japan | Vienna | St | Schengen |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Et | Si | Lëtzebuerg | Dat | Se |
|  | En | [MASK] is the Capital of France. | it | it | Luxembourg | that | it |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Chef | Premier | Mann | Patron | Kinnek |
|  | En | Xavier Bëttel is the [MASK] of Luxembourg. | boss | prime minister | man | owner | king |

(b) Suggestions for Lb BERT *cased*

Table 6.9: Performance of the uncased and cased Lb/De BERT models. Suggestions to replace the [MASK] tokens are ordered by confidence score. We highlighted in green whether a suggestion is valid.

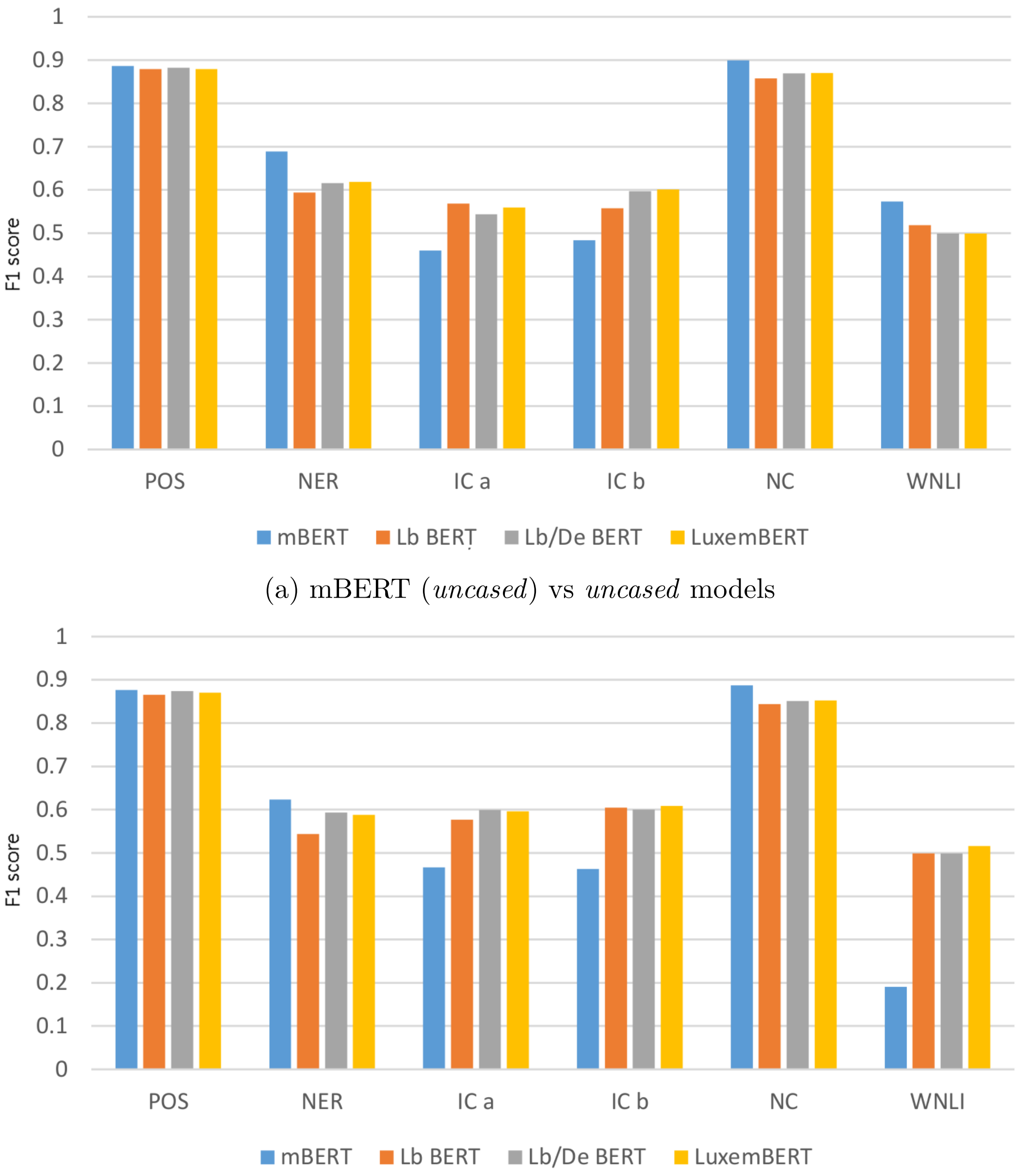
| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | ginn | gin | muss | war | fueren |
|  | En | I [MASK] to school. | go | go | have to (go to) | was | drive |
| 2 | Lb | Ech ginn an d'[MASK]. | Vakanz | F | Pensioun | Schoul | Mass |
|  | En | I go to the [MASK]. | holiday | f | retirement | school | church mass |
| 3 | Lb | Ech iessen en [MASK]. | net | o | h | . | z |
|  | En | I eat a/it [MASK]. | not | o | h | . | z |
| 4 | Lb | Ech iessen e [MASK]. | g | k | p | s | w |
|  | En | I eat a [MASK]. | g | k | p | s | w |
| 5 | Lb | Ech iessen eng [MASK]. | s | k | g | b | z |
|  | En | I eat a [MASK]. | s | k | g | b |  |
| 6 | Lb | Den Hond läit um [MASK]. | Réck | Bauch | Kapp | Ouer | Hals |
|  | En | The dog lies on the/its [MASK]. | back | belly | head | ear | neck |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Diddeleng | Beetebuerg | Lëtzebuerg | Ettelbréck | Wolz |
|  | En | The (motorway) A4 connects Esch and [MASK]. | Dudelange | Bettembourg | Luxembourg | Ettelbruck | Wiltz |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Paräis | Frankräich | Bréissel | Marseille | Lëtzebuerg |
|  | En | Paris is the Capital of [MASK]. | Paris | France | Brussels | Marseille | Luxembourg |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Et | Si | Stroossbuerg | Paräis | Lëtzebuerg |
|  | En | [MASK] is the Capital of France. | it | it | Strasbourg | Paris | Luxembourg |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Buergermeeschter | Premier | Premierminister | President | Staatsminister |
|  | En | Xavier Bëttel is the [MASK] of Luxembourg. | mayor | prime minister | prime minister | president | minister of state |

(a) Suggestions for De/Lb BERT *uncased*

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | ginn | gin | war | muss | fueren |
|  | En | I [MASK] to school. | go | go | was | have to (go to) | drive |
| 2 | Lb | Ech ginn an d'[MASK]. | m | k | Schoul | d | f |
|  | En | I go to the [MASK]. | m | k | school | d | f |
| 3 | Lb | Ech iessen en [MASK]. | a | i | t | z | d |
|  | En | I eat a/it [MASK]. | a | i | t | z | d |
| 4 | Lb | Ech iessen e [MASK]. | k | s | g | l | b |
|  | En | I eat a [MASK]. | k | s | g | l | b |
| 5 | Lb | Ech iessen eng [MASK]. | s | k | p | b | r |
|  | En | I eat a [MASK]. | s | k | p | b | r |
| 6 | Lb | Den Hond läit um [MASK]. | Kapp | a | Mo | Been | h |
|  | En | The dog lies on the/its [MASK]. | head | a | stomach | leg | h |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Lëtzebuerg | St | Stad | Ell | Käl |
|  | En | The (motorway) A4 connects Esch and [MASK]. | Luxembourg | St | city | Ell | Kayl |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Lëtzebuerg | Japan | Berlin | Wien | Europa |
|  | En | Paris is the Capital of [MASK]. | Luxembourg | Japan | Berlin | Vienna | Europe |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Et | Si | Lëtzebuerg | Dat | Se |
|  | En | [MASK] is the Capital of France. | it | it | Luxembourg | that | it |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Chef | Premier | Mann | Patron | Minister |
|  | En | Xavier Bëttel is the [MASK] of Luxembourg. | boss | prime minister | man | owner | minister |

(b) Suggestions for De/Lb BERT *cased*

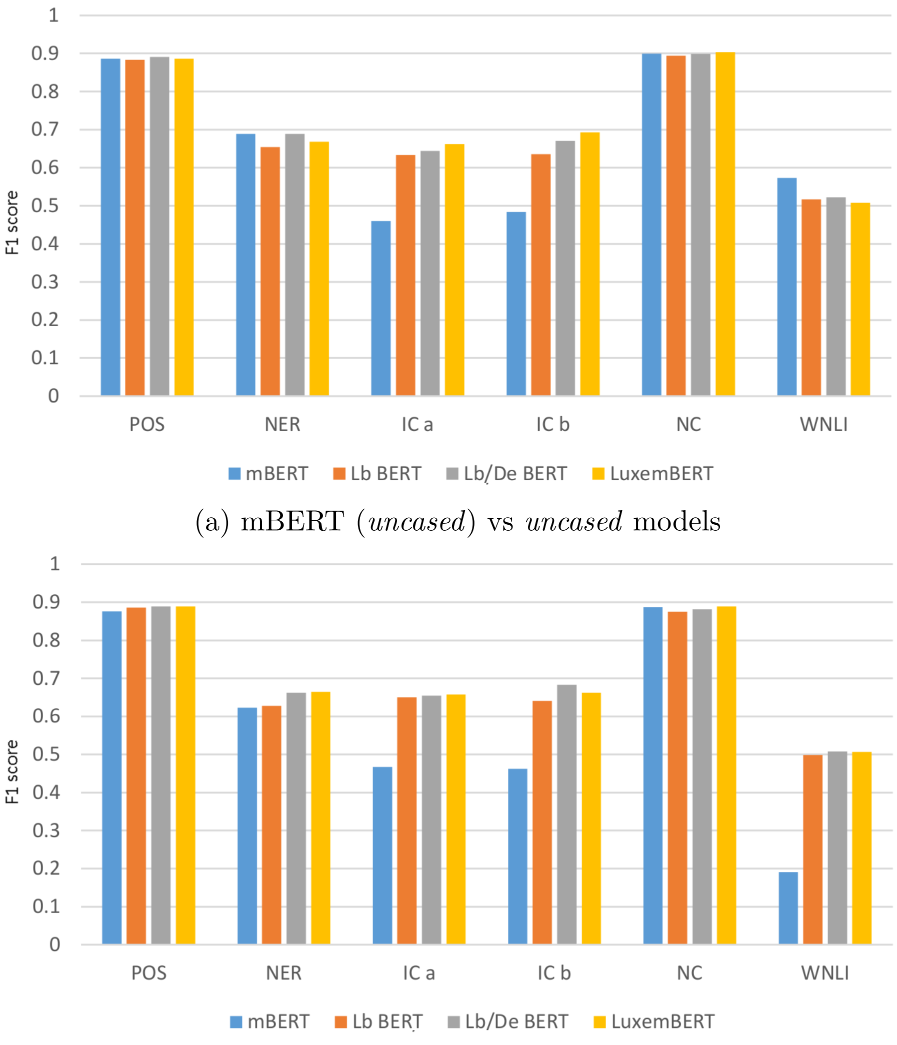
In addition to leveraging the presence of German data, we also leverage the size of the pre-training corpus to quantify how much adding the auxiliary language can improve a language model in the case where the lack of data is even more apparent. Figures 6.5, 6.6, and 6.7 show the performances of our models trained on *small*, *medium* and *large* datasets, respectively. The results of the F/N test can be found in Figures 6.8a to 6.8f. 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 exceptions: (1) for *small* uncased, mBERT seems to be more competitive, and (2) for *large*, Lb BERT is in-between Lb/De BERT and LuxemBERT.



(a) mBERT (*uncased*) vs *uncased* models

(b) mBERT (*cased*) vs *cased* models

Figure 6.5: Comparison of Lb BERT, Lb/De BERT, and LuxemBERT to mBERT on the small-sized dataset



(a) mBERT (*uncased*) vs *uncased* models

(b) mBERT (*cased*) vs *cased* models

Figure 6.6: Comparison of Lb BERT, Lb/De BERT, and LuxemBERT to mBERT on the medium-sized dataset

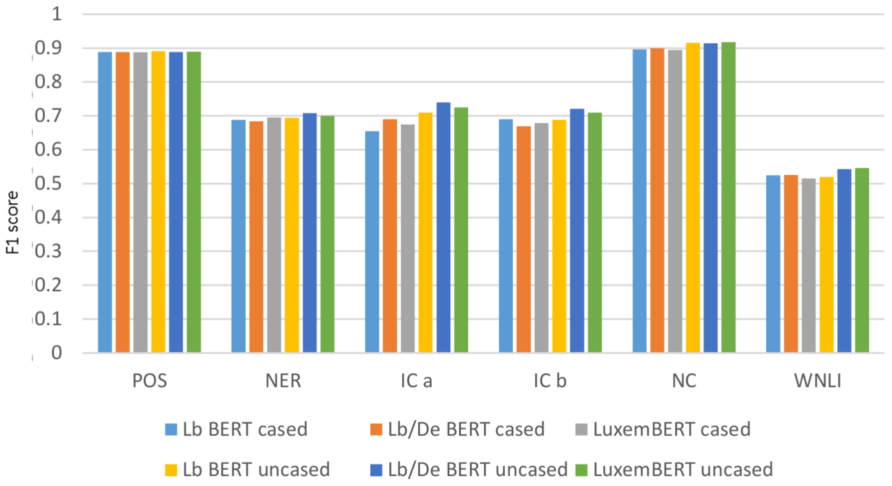
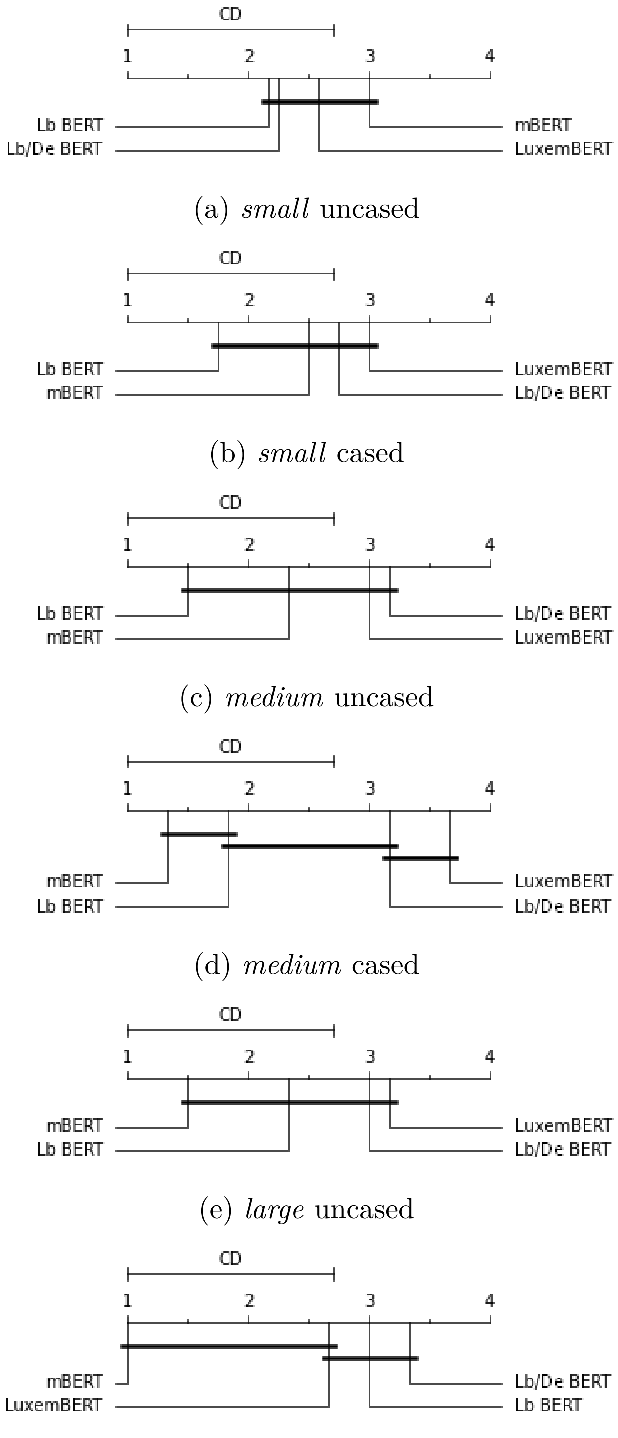


Figure 6.7: Comparison of Lb BERT, Lb/De BERT, LuxemBERT on large dataset



(a) *small* uncased

(b) *small* cased

(c) *medium* uncased

(d) *medium* cased

(e) *large* uncased

(f) *large* cased

Figure 6.8: 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 6.10.

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.

## 6.5 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.

Table 6.10: Wins/ties/losses comparison, based on Wilcoxon superiority tests, of all models of this study.

| 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 |

As a last consideration, we compare all variants to search for the best among all 20 alternatives presented in this chapter. 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 6.10. 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.

## 6.6 Related Work

### 6.6.1 Training Language Models on Multilingual Data

As mentioned in Section 2.1.3.4, multilingual language models such as mBERT serve as an important language model for numerous less widespread languages as it offers versatility at the expense of performance. Indeed, Wu et al. [43] 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.

In addition, Wu et al. 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.

## 6.7 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.

## 6.8 Summary

In this chapter, 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.

# 7 Comparing Pre-Training Schemes for Luxembourgish BERT Models

*In this chapter, we propose two novel Luxembourgish BERT models 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.*

This chapter is based on the work in the following research paper under submission:

- Cedric Lothritz, Saad Ezzini, Christoph Purschke, Tegawendé F. Bissyandé, Jacques Klein, Andrey Boytsov, Clément Lefebvre, Anne Goujon, and Isabella Olariu. Comparing Pre-Training Schemes for Luxembourgish BERT Models, *Konferenz zur Verarbeitung natürlicher Sprache*, 2023

## Contents

## 7.1 Overview

The introduction of BERT models in 2018 [3] (cf. Section 2.1.3.2) 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 mBERT [3] and XLM-RoBERTa [119] (cf. Section 2.1.3.3). These multilingual models provide a viable alternative, but monolingual models can outperform them if sufficient pre-training data is available, as shown by Wu et al.[43].

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 [120] (cf. Chapter 6). The configuration of the model architecture can also be varied to improve performance, as highlighted by Wu et al.[43]. Another approach 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 [121]. 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. 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) Two novel BERT models for the Luxembourgish language that improve on the state of the art

(b) An empirical study on both the performance and robustness of the investigated BERT models

(c) Novel datasets to evaluate the performance of Luxembourgish language models

The remainder of this chapter is structured as follows: Section 7.2 describes our approach to building our novel language models. We establish our research questions, and describe our overall experimental setup for this study in Section 7.3. In Section 7.4, we present the results from our experiments, which we then discuss in Section 7.5. Section 7.6 shows a selection of works that are related to our own study. We lay down a number of potential threats to the veracity of our experiments in Section 7.7. Finally, we conclude our findings in Section 7.8.

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

## 7.2 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.

### 7.2.1 Pre-loaded Models

As mentioned in Section7.1, 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].

#### 7.2.1.1 mBERT

As mentioned before, the mBERT model was pre-trained on Wikipedia articles, including the Luxembourgish Wikipedia, which contained 59 000 articles at the time of training. 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%.

#### 7.2.1.2 GottBERT

Luxembourgish is a West Germanic language originating from a Moselle Franconian dialect [8]. As such, Luxembourgish and German are closely related. As mentioned in Section 6.2, both languages are similar in terms of vocabulary and structure. Due to these similarities, we choose the German GottBERT model [41] as a pre-loaded model to create Lb_GottBERT. GottBERT was pre-trained on the German part of the OSCAR corpus [38] 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.

### 7.2.2 Pre-training Corpus

In order to pre-train our models, we use the same corpus that we used to build LuxemBERT (cf. Section 6.2.1) which consists of 12 million sentences, 6 million of which are written in Luxembourgish.

## 7.3 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 fine-tuned 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

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

(RTE) (cf. Section 2.2). Furthermore, when applicable, we apply four perturbation techniques to our test sets: negation, name replacement, location replacement, and synonym replacement.

## 7.3.1 Research Questions

We address the following two research questions:

1. 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, POS-tagging, etc. The goal is to identify the model that performs the best across all tasks or a specific set of tasks.

2. 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.

## 7.3.2 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 7.1 shows an overview of the differences between each model.

Table 7.1: Differences in pre-training scheme and data for each investigated model. (NAP = no additional pre-training)

| | 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 |

### 7.3.2.1 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.

### 7.3.2.2 LuxemBERT

We reuse the LuxemBERT model (cf. Chapter 6) that we made from scratch trained on the 12 million sentences described in Section 6.2.1. 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%. As LuxemBERT already outperforms mBERT in numerous tasks, we expect it to outperform both mBERT and GottBERT in most of our experiments.

#### 7.3.2.3 DA BERT

DA BERT was created by Olariu et al.[122] 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.

### 7.3.3 Downstream Tasks

For this study, we consider eight downstream tasks. In addition to the five tasks introduced in Section 6.3.3 (POS-tagging, Named Entity Recognition, Intent Classification, News Classification, and WNLI), we also investigate Sentence Negation, Sentiment Analysis, and the Recognizing Textual Entailment task, which we describe in the following section[3].

#### 7.3.3.1 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 [45][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%).

#### 7.3.3.2 Recognizing Textual Entailment

The Recognizing Textual Entailment (RTE) task was introduced by Haim et al. [60] and was added to the GLUE benchmark collection [30] for evaluating the performance of language models (cf. Section 2.2.3.1. 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, but it is serviceable for our experiments. 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] 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/

#### 7.3.3.3 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 [26] 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.

### 7.3.4 Fine-tuning Parameters

Devlin et al. [3] recommends choosing hyperparameters for batch size, learning rate, and number of training epochs from the following ranges: $range_{batch\,size}=\{16,32\}$, $range_{learning\,rate}=\{2e\text{-}5, 3e\text{-}5, 5e\text{-}5\}$, and $range_{epochs}=\{1,2,3,4,5\}$. For the POS, NER, IC, NC, and WNLI tasks, we reuse the same parameters from Section 6.3.4, for the remaining tasks, we perform a grid search using the original LuxemBERT model to find the best-performing configuration of parameters. Table 7.2 shows the chosen hyperparameters for each task. We fine-tune each of our models on the same sets of hyperparameters.

Table 7.2: Fine-tuning hyperparameters for each investigated task

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

### 7.3.5 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. [123]: 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 7.3 shows an overview of the techniques we use.

#### 7.3.5.1 Negation

As described in Section 7.3.3.1, 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

[7] www.rtl.lu

[8] We make this dataset available on request

Table 7.3: Applicability of the perturbation techniques

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

one sentence of each *entailment* pair in the test set. This approach will turn an *entailment* sentence pair into a *not_entailment* pair.

#### 7.3.5.2 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. In order to maintain consistency, we ensure that identical names in the datasets are all mapped to the same names during the replacement.

#### 7.3.5.3 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

## 7.4 Experimental Results

In this section, we will present the detailed results from our experiments. Table 7.4 shows the average performance of each model on each task using the original test sets in terms of F1 score. Table 7.5 displays the performances on original and perturbed test sets of each model fine-tuned on Sentence Negation, RTE, and Sentiment Analysis.

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

Similarly to Section 6.4, we first test the performance of each model on the MLM task using the 10 example sentences introduced in Section 6.2.2. Tables 6.2, 7.6, 7.7, and 7.8 show the suggestions to each example sentence from each of the investigated models. Overall, it appears that the original LuxemBERT model still performs best with all suggestions being grammatically correct and most suggestions making sense from a semantic standpoint (cf. Table 6.2). Notable exceptions are the suggestions related to food (sentences 3, 4, and 5) where the model oftentimes suggests drinks rather than foods. However, it is to note that every other model struggles with predictions for these particular sentences, most of them suggesting single letters rather

Table 7.4: Results for each task on the original test sets. * denotes naive classifier that always predicts the same class

| 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 7.5: 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)

| 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 |

than words. Both DA BERT (cf. Table 7.6) and Lb_GottBERT (cf. Table 7.8b) perform similarly well to each other, only performing poorly when tested on sentences 3, 4, and 5. Lb_mBERT (cf. Table 7.7b) additionally struggles with sentence 2, mostly suggesting letters instead of words. The unmodified mBERT model (cf. Table 7.7a) performs well with sentences 7, 8, 9, and 10. As these sentences can be regarded as statements of fact commonly found in documents such as encyclopaedias, it is understandable that mBERT performs well with these types of sentences as it was pre-trained exclusively on Wikipedia articles. GottBERT (cf. Table 7.8a), which was not trained on any Luxembourgish data at all, performs worst in this task. The model tends to either suggest seemingly random German words, or add suffixes to the word before the masking token in order to create German words. For example, regarding sentence 6, GottBERT only suggests verbs which, when added to the word "um" result in new verbs.

Table 7.6: Suggestions for DA BERT

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | gin | ginn | muss | war | wëll |
| | En | I [MASK] to school. | go | go | have to (go to) | was | want (to go to) |
| 2 | Lb | Ech ginn an d'[MASK]. | Vakanz | Schoul | Congé | Ausland | f |
| | En | I go to the [MASK]. | holiday | school | annual leave | foreign countries | f |
| 3 | Lb | Ech iessen en [MASK]. | a | z | net | och | o |
| | En | I eat a/it [MASK]. | a | z | not | also | o |
| 4 | Lb | Ech iessen e [MASK]. | k | b | s | r | l |
| | En | I eat a [MASK]. | k | b | s | r | l |
| 5 | Lb | Ech iessen eng [MASK]. | s | k | b | a | z |
| | En | I eat a [MASK]. | s | k | b | a | z |
| 6 | Lb | Den Hond läit um [MASK]. | Buedem | Réck | Schwanz | Kapp | Bauch |
| | En | The dog lies on the/its [MASK]. | floor | back | tail | head | belly |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Diddeleng | Esch | Déifferdeng | Beetebuerg | Schëffleng |
| | En | The (motorway) A4 connects Esch and [MASK]. | Dudelange | Esch | Differdange | Bettembourg | Schifflange |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Frankräich | Paräis | Lëtzebuerg | Spuenien | Katalounien |
| | En | Paris is the Capital of [MASK]. | France | Paris | Luxembourg | Spain | Catalonia |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Paräis | Et | Si | Lille | Bréissel |
| | En | [MASK] is the Capital of France. | Pari | sit | it | Lille | Brussels |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Premier | Buergermeeschter | Premierminister | Staatsminister | President |
| | En | Xavier Bëttel is the [MASK] of Luxembourg. | prime minister | mayor | prime minister | minister of state | president |

Table 7.7: Performance of MLM task on the mBERT models

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | ##ill | ##uel | ##ed | ##eh | ##ap |
| | En | I [MASK] to school. | ##ill | ##uel | ##ed | ##eh | ##ap |
| 2 | Lb | Ech ginn an d'[MASK]. | Stad | u | z | 2 | nr |
| | En | I go to the [MASK]. | town | u | z | 2 | nr |
| 3 | Lb | Ech iessen en [MASK]. | nederland | Europa | zee | belgie | Vlaanderen |
| | En | I eat a/it [MASK]. | Netherlands | Europe | sea | Belgium | Flanders |
| 4 | Lb | Ech iessen e [MASK]. | ##w | s | al | ! | man |
| | En | I eat a [MASK].##w | s | old | ! | man | |
| 5 | Lb | Ech iessen eng [MASK]. | ##r | ##h | ##s | ##l | ##g |
| | En | I eat a [MASK]. | ##r | ##h | ##s | ##l | ##g |
| 6 | Lb | Den Hond läit um [MASK]. | km$^2$ | m | km | hektar | Island |
| | En | The dog lies on the/its [MASK]. | km$^2$ | m | km | hektar | Iceland |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Lëtzebuerg | Köln | Luxemburg | Aachen | Koblenz |
| | En | The (motorway) A4 connects Esch and [MASK]. | Luxembourg | Cologne | Luxembourg | Aachen | Koblenz |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Lëtzebuerg | frans | paris | Frankreich | france |
| | En | Paris is the Capital of [MASK]. | Luxembourg | French | Paris | France | France |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Metz | Nancy | Toulouse | Troyes | Poitiers |
| | En | [MASK] is the Capital of France. | Metz | Nancy | Toulouse | Troyes | Poitiers |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Politiker | President | Gouverneur | maire | Premierminister |
| | En | Xavier Bëttel is the [MASK] of Luxembourg. | politician | president | governor | mayor | prime minister |

(a) Suggestions for mBERT

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | ginn | gin | muss | war | fueren |
| | En | I [MASK] to school. | go | go | have to (got to) | was | drive |
| 2 | Lb | Ech ginn an d'[MASK]. | t | Pensioun | f | a | l |
| | En | I go to the [MASK]. | t | retirement | f | a | l |
| 3 | Lb | Ech iessen en [MASK]. | d | z | o | s | . |
| | En | I eat a/it [MASK]. | d | z | o | s | . |
| 4 | Lb | Ech iessen e [MASK]. | g | k | b | s | p |
| | En | I eat a [MASK]. | g | k | b | s | p |
| 5 | Lb | Ech iessen eng [MASK]. | z | p | s | k | d |
| | En | I eat a [MASK]. | z | p | s | k | d |
| 6 | Lb | Den Hond läit um [MASK]. | i | Buedem | Hals | Bauch | t |
| | En | The dog lies on the/its [MASK]. | i | floor | neck | belly | t |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Beetebuerg | Péiteng | Schëffleng | Diddeleng | Sussem |
| | En | The (motorway) A4 connects Esch and [MASK]. | Bettembourg | Pétange | Schifflange | Dudelange | Sanem |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Frankräich | Paräis | Spuenien | Lëtzebuerg | Frankreich |
| | En | Paris is the Capital of [MASK]. | France | Paris | Spain | Luxembourg | France |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Et | Paräis | Versailles | Lille | Metz |
| | En | [MASK] is the Capital of France. | it | Paris | Versailles | Lille | Metz |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Premier | Premierminister | Buergermeeschter | Regierungsminister | Staatsminister |
| | En | Xavier Bëttel is the [MASK] of Luxembourg. | prime minister | prime minister | mayor | government minister | minister of state |

(b) Suggestions for Lb_mBERT

Table 7.8: Performance of MLM task on the GottBERT models

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | nung | os | se | tel | . |
| | En | I [MASK] to school. | nung | os | se | tel | . |
| 2 | Lb | Ech ginn an d'[MASK]. | er | frau | mann | männer | Freunde |
| | En | I go to the [MASK]. | he | man | woman | men | friends |
| 3 | Lb | Ech iessen en [MASK]. | Youtube | Pinterest | Anleitung | kaufen | Balkon |
| | En | I eat a/it [MASK]. | Youtube | Pinterest | manual | buy | balcony |
| 4 | Lb | Ech iessen e [MASK]. | ch | den | mail | ct | de |
| | En | I eat a [MASK]. | ch | den | mail | ct | de |
| 5 | Lb | Ech iessen eng [MASK]. | land | ine | ingen | kaufen | agement |
| | En | I eat a [MASK]. | country | ine | ingen | buy | agement |
| 6 | Lb | Den Hond läit um [MASK]. | rühren | füllen | drehen | gießen | legen |
| | En | The dog lies on the/its [MASK]. | stir | fill | turn | water | lay |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | 50 | 60 | 70 | 40 | 94 |
| | En | The (motorway) A4 connects Esch and [MASK]. | 50 | 60 | 70 | 40 | 94 |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | est | es | j | vert | o |
| | En | Paris is the Capital of [MASK]. | est | es | j | vert | o |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | ass | , | and | ass | the |
| | En | [MASK] is the Capital of France. | ass | , | and | ass | the |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | la | france | e | de | mer |
| | En | Xavier Bëttel is the [MASK] of Luxembourg. | la | france | e | de | mer |

(a) Suggestions for GottBERT

| ID | Lang | Sentence | Suggestion 1 | Suggestion 2 | Suggestion 3 | Suggestion 4 | Suggestion 5 |
|---|---|---|---|---|---|---|---|
| 1 | Lb | Ech [MASK] an d'Schoul. | ginn | gin | muss | war | fueren |
| | En | I [MASK] to school. | go | go | have to (got to) | was | drive |
| 2 | Lb | Ech ginn an d'[MASK]. | Vakanz | Pensioun | Kierch | Schoul | Ausland |
| | En | I go to the [MASK]. | holiday | retirement | church | school | foreign countries |
| 3 | Lb | Ech iessen en [MASK]. | h | a | o | z | e |
| | En | I eat a/it [MASK]. | h | a | o | z | e |
| 4 | Lb | Ech iessen e [MASK]. | g | k | s | f | p |
| | En | I eat a [MASK]. | g | k | s | f | p |
| 5 | Lb | Ech iessen eng [MASK]. | s | k | f | p | z |
| | En | I eat a [MASK]. | s | k | f | p | z |
| 6 | Lb | Den Hond läit um [MASK]. | Buedem | Schwanz | Réck | Bauch | Kapp |
| | En | The dog lies on the/its [MASK]. | floor | tail | back | belly | head |
| 7 | Lb | D'A4 verleeft vun Esch bis op [MASK]. | Beetebuerg | Belval | Péiteng | Tréier | Diddeleng |
| | En | The (motorway) A4 connects Esch and [MASK]. | Bettembourg | Belval | Pétange | Troyes | Dudelange |
| 8 | Lb | Paräis ass d'Haaptstad vu [MASK]. | Paräis | Frankräich | Marseille | Lëtzebuerg | Bréissel |
| | En | Paris is the Capital of [MASK]. | Paris | France | Marseille | Luxembourg | Brussels |
| 9 | Lb | [MASK] ass d'Haaptstad vu Frankräich. | Paräis | Metz | Et | Marseille | Versailles |
| | En | [MASK] is the Capital of France. | Paris | Metz | it | Marseille | Versailles |
| 10 | Lb | De Xavier Bëttel ass de [MASK] vu Lëtzebuerg. | Premier | Buergermeeschter | Premierminister | President | Spëtzekandidat |
| | En | Xavier Bëttel is the [MASK] of Luxembourg. | prime minister | mayor | prime minister | president | prime candidate |

(b) Suggestions for Lb_GottBERT

For a more in-depth answer to this question, we refer to the results shown in both Table 7.4 and Figure 7.1. 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.

**RQ1 Answer:** The models with no additional pre-training generally performed worst, the models that were pre-trained with pre-loaded models performed best, and the models pre-trained from scratch yielded intermediate results. Overall, Lb_GottBERT generally performed best.

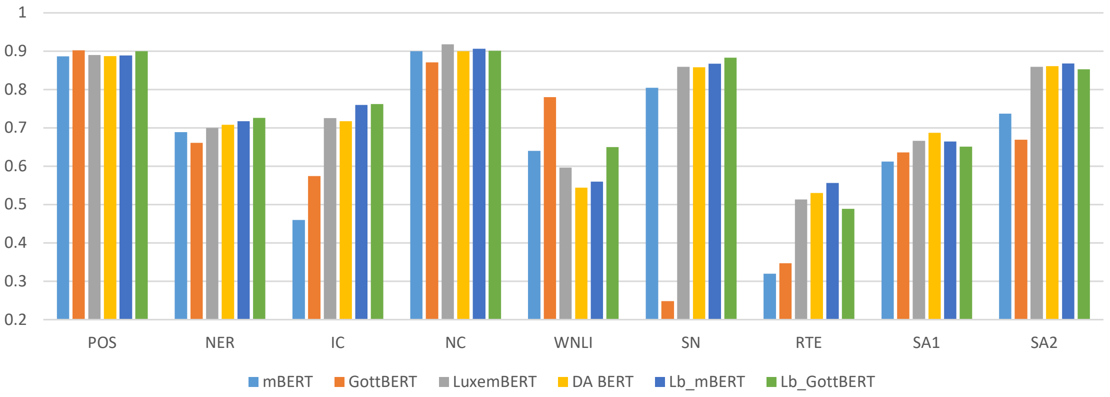


Figure 7.1: Fine-tuning results of the models on each investigated task

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

In order to answer this question, we applied the perturbation techniques as described in Section 7.3.5 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 7.5.

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. Figures 7.2, 7.3, 7.4, 7.5)

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 the same answer: *3*. As such, this score is not meaningful.

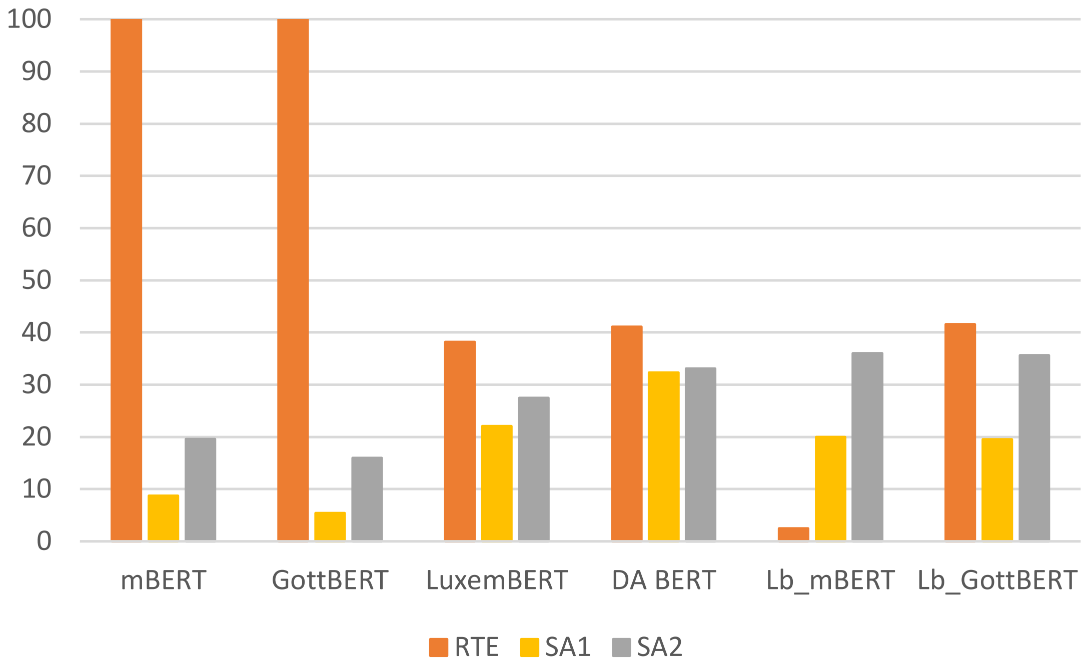


Figure 7.2: Impact of negation on each model's performance.

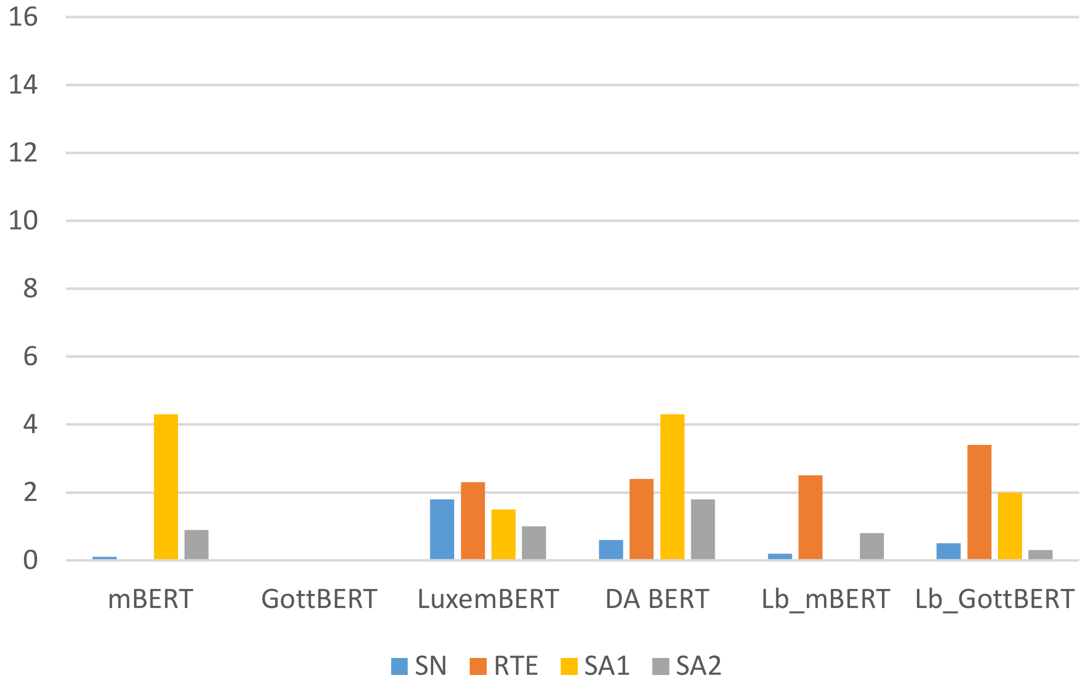


Figure 7.3: Impact of name replacements on each model's performance.

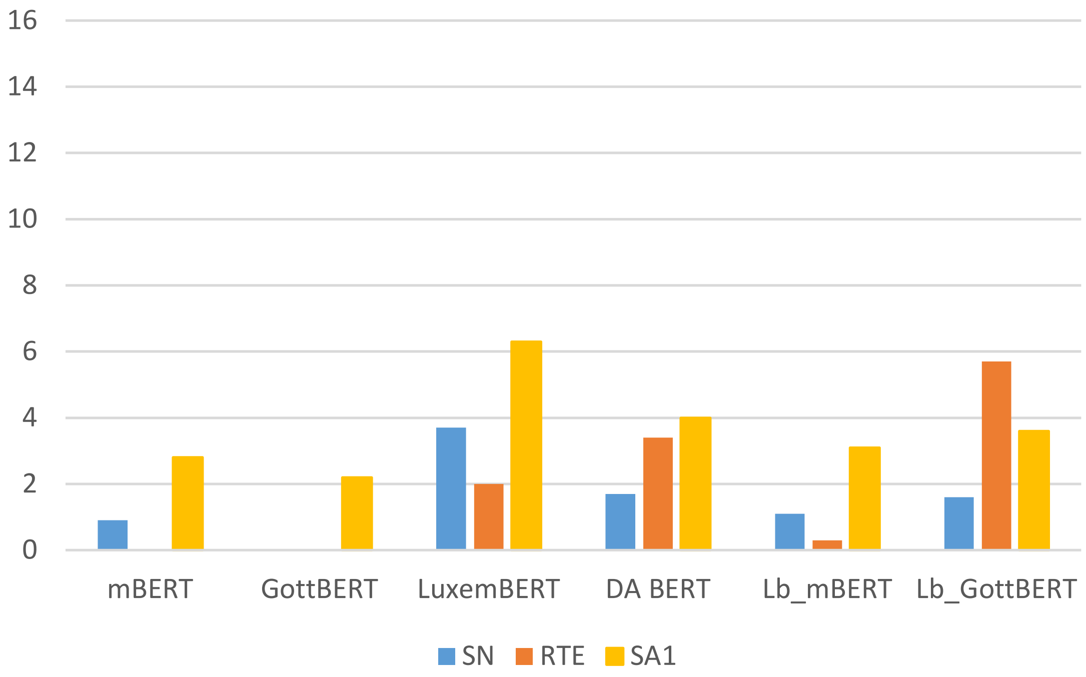


Figure 7.4: Impact of location replacements on each model's performance.

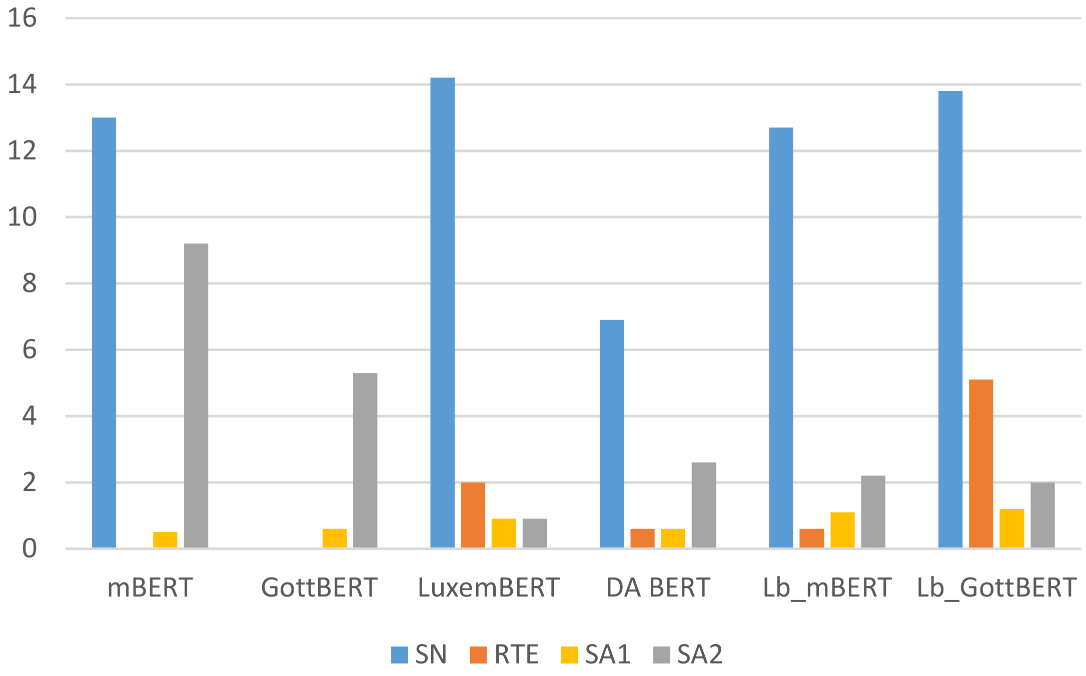


Figure 7.5: Impact of synonym replacements on each model's performance.

While the differences are very low for entity replacements, we notice significant differences for synonym replacement, most of which are close to 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.

**RQ2 Answer:** Most models were highly affected by sentence negation, moderately affected by synonym replacement, and barely impacted by both name replacement or location replacement. Lb_mBERT was shown to be the most robust of our models overall.

## 7.5 Discussion

Overall, both Lb_mBERT and Lb_GottBERT outperform LuxemBERT and DA BERT in almost all tasks. (cf. Table 7.4) 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.

## 7.6 Related Works

### 7.6.1 Pre-Training Pre-Loaded Language Models

Similar to our approach, Muller et al. [121] 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.

### 7.6.2 Evaluating the Robustness of Language Models

Ribeiro et al. [123] 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.

Most of our own data perturbation techniques were inspired by data augmentation approaches for low-resource languages detailed by Hedderich et al. [120] These techniques included synonym replacements and named entity replacement such as location replacement.

The ability of BERT models to handle negation on the task of Sentiment Analysis was studied by Tejada et al. [124] who found that these models were not truly capable of handling the concept of negation. Furthermore, models that were not shown any negated sentences during the training performed poorly on negated sentences. A similar result was found in our own study.

## 7.7 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 7.2.1. 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.

## 7.8 Summary

In this chapter, 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.

# 8 Conclusion

*In this chapter, we will summarise our contributions, and provide an outlook into the future by discussing various ideas for follow-up research directions.*

## Contents

## 8.1 Summary

In this thesis, we discussed various domain-specific and language-specific NLP challenges that arise in a multilingual country such as Luxembourg. We emphasised three key aspects to be considered when handling NLP systems in Luxembourg: challenges related to the financial domain, challenges related to multilingualism, and challenges related to the Luxembourgish language. In short, our contributions to the NLP community include multiple empirical studies, three novel language models, and nine novel datasets for various NLP tasks.

With regard to NLP challenges relevant to the banking domain, we focused on handling names in documents, but also generalised the task to include non-financial documents as well. In Chapter 3, we performed an empirical study on the performance of models based on the Transformer architectures BERT, RoBERTa, and XLNet on the Fine-Grained Named Entity Recognition task, and compared them to two non-Transformer based models: a simple CRF and an ensemble BiLSTM-CNN-CRF model. We performed the study on 49 different domains including the banking domain, the legal domain, and scientific domains. We determined that Transformer-based models indeed outperform the competition in this task in terms of recall and F1 score, but not in terms of precision. Specifically, the simple BERT model performed best in 36 out of 49 domains and RoBERTa in 10 out of 49 in terms of f1 score. We furthermore noticed that there is a high correlation between the performance of a model and the choice of the domain which could not be explained by the size of the respective datasets. We then investigated in Chapter 4 how much the performance of NLP models is impacted after the training datasets have been anonymised. We considered two Transformer-based models, BERT and ERNIE, and six anonymisation strategies applied to datasets for nine NLP tasks varying in difficulty. We found that the impact of anonymisation before model training exists, but is relatively low. We determined that the best results were achieved when anonymising using random names. We furthermore found and recommend to anonymise the data prior the model training to increase the performance gain of fine-tuned models.

Addressing challenges related to multilingualism, we performed an empirical study on multilingual chatbots for the banking domain. Specifically, in Chapter 5 we targeted the NLU capabilities of a chatbot, i.e. the Intent Classification and Slot Filling tasks when trained on up to four languages. We concluded that chatbots trained on mixed-language datasets lead to a worse performance, and that the decrease in performance correlated with the increase in number of languages. We also found that training several chatbots on separate datasets is usually preferable to training a single chatbot on a mixed-language dataset. In addition to the study, we also publish a novel multilingual dataset for Intent Classification and Slot Filling.

With respect to Luxembourgish NLP, we focused on challenges related to the training of BERT models for the Luxembourgish language. In Chapter 6, we introduced LuxemBERT as the first Transformer-based model for Luxembourgish. We showed that our data augmentation technique is viable to create new textual data for language model pre-training. We also show that our LuxemBERT outperforms its sole competitor mBERT in five out of six downstream NLP tasks. Additionally, we make the model itself as well as four NLP datasets in Luxembourgish available to the community. Finally, in Chapter 7, we investigated various strategies to pre-train BERT models. We specifically perform a study to determine whether it is preferable to pre-train a model from scratch, or to continue pre-training using an already existing model. Based on the approach of continued pre-training from a pre-loaded model, we trained and published two additional Luxembourgish language models named Lb_mBERT and Lb_GottBERT. Furthermore, we introduced and published four additional Luxembourgish datasets for NLP tasks. We found that our LuxemBERT model from Chapter 6 also outperforms mBERT in these four additional tasks, increasing the number from five out of six to nine of ten. We, furthermore found that our two novel language models further advance the state of the art as Lb_mBERT performs best in two out of nine investigated tasks and Lb_GottBERT performs best in four out of nine tasks. We additionally performed an evaluation of the robustness of each of our models and found that the performance of our models was highly impacted by sentence negation. Simple word replacements on the other hand had a low to moderate effect on the performance. We found that Lb_mBERT was typically the most robust out of all the models.

## 8.2 Future Work

Finally, we present multiple potential research opportunities to expand or improve on the work featured in this dissertation:

- *Multilingual Fine-Grained Named Entity Recognition*: Our findings from Chapter 3 did show that Transformer-based models perform well on this task. However, we only considered English data in our study. As there is also a Turkish version of the EWNERTC dataset that we used, it would be interesting to investigate how well Turkish models such as BERTurk [125], and multilingual models such as mBERT [3] or XLM-RoBERTa [119] would perform in this task. We would also be interested in addressing other languages, but that would require significant effort as these kinds of datasets are scarce and require sizeable effort to create. Finally, we would like to determine to what extent the choice of domain impacts the model performance in other languages.
- *Applying Domain-specific Models to Fine-Grained Named Entity Recognition* For our study on FGNER from Chapter 3, we used the general-purpose BERT, RoBERTa, and XLNET models which were trained on broad corpora. However, some of the subsets that were featured in that study contain many domain-specific words that do not necessarily appear in those corpora. As such, we suggest to investigate the performance of domain-specific models such as FinBERT or SciBERT(cf. Section 2.1.3.4) applied to appropriate subsets.
- *Translation-based Chatbots*: In our study regarding multilingual chatbots in Chapter 5, we considered two implementation strategies: monolingual chatbots combined with a language selector, and a single chatbot trained on multilingual data. As a third strategy, we can consider a combination of a monolingual and a translation tool such as the recently released NLLB-model [126]. For instance, we found that the English chatbot we trained yielded the highest performance. Depending on the quality of the translations, this strategy might lead to better results than our own approaches. In addition, as there was no BERT model for the Luxembourgish language available at the time of the study, we decided to use mBERT to train the monolingual chatbots. However, with the release of

our LuxemBERT model, we could train them with language-specific language models instead to increase the performance of these models.

- *Improvement of LuxemBERT*: We already trained and presented multiple Luxembourgish BERT models in Chapters 6 and 7, all of which perform well on a variety of NLP tasks. However, while we considered the close relationship between the German and the Luxembourgish languages when augmenting, we did not consider the influence of the French language on Luxembourgish. A large number of Luxembourgish words have their roots in the French language, in particular nouns, adjectives, and verbs. In addition to translate function words from German to Luxembourgish, we could translate certain content words from German to French, resulting in a new pre-training corpus for a Luxembourgish language model. We would be interested in determining if this new augmentation technique would result in further performance increases.
- *Generalise Our Data Augmentation Method to any Low-Resource Language*: In Chapter 6, we showed that our data augmentation scheme is to an extent useful in the case of the Luxembourgish language. It would be interesting to apply this technique to other language pairs to train low-resource languages. Obvious examples include Latvian/Lithuanian to train a Latvian BERT model or Afrikaans/Dutch for an Afrikaans model, which are the language pairs investigated by Wu et al. [43]. We can broaden our scope to build a repository that includes numerous low-resource and endangered languages. This would be an arduous, yet valuable undertaking in an effort to preserve these languages.
- *Domain-specific Language Models in a Low-resource or Multilingual Setting*: In this dissertation, we found that there is a general lack of language models for low-resource languages as well as differences in performance when applying a general-purpose language model to different domains. Following these findings, we can mitigate these issues by building either multilingual or low-resource language models trained on domain-specific data. The data augmentation technique we used for creating Luxembourgish data can be expanded by including domain-specific vocabulary in our translation scheme. For instance, we could apply this modified data augmentation technique to financial textual data written in a high-resource language to create adequate data for a low-resource language. This way, we could complement already existing models such as FinBERT for the financial domain, significantly benefiting the NLP community at large.

# List of Papers

**Papers included in this dissertation:**

- Cedric Lothritz, Kevin Allix, Lisa Veiber, Tegawendé F. Bissyandé, Jacques Klein, Evaluating pretrained transformer-based models on the task of fine-grained named entity recognition, *Proceedings of the 28th International Conference on Computational Linguistics*, 2020
- Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Tegawendé F. Bissyandé, Jacques Klein, Andrey Boytsov, Clément Lefebvre, and Anne Goujon. Evaluating the Impact of Anonymisation on Downstream NLP Tasks, *Nordic Conference on Computational Linguistics*, 2023
- Cedric Lothritz, Kevin Allix, Bertrand Lebichot, Lisa Veiber, Tegawendé F. Bissyandé, Jacques Klein. Comparing MultiLingual and Multiple MonoLingual Models for Intent Classification and Slot Filling, *International Conference on Applications of Natural Language to Information Systems*, 2021
- Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Lisa Veiber, Tegawendé F. Bissyandé, Jacques Klein, Andrey Boytsov, Clément Lefebvre, and Anne Goujon. LuxemBERT: Simple and Practical Data Augmentation in Language Model Pre-Training for Luxembourgish, *Proceedings of the Language Resources and Evaluation Conference*, 2022
- Cedric Lothritz, Saad Ezzini, Christoph Purschke, Tegawendé F. Bissyandé, Jacques Klein, Andrey Boytsov, Clément Lefebvre, Anne Goujon, and Isabella Olariu. Comparing Pre-Training Schemes for Luxembourgish BERT Models, To be submitted to *Konferenz zur Verarbeitung natürlicher Sprache*, 2023

**Papers not included in this dissertation:**

- Yusuf Arslan, Kevin Allix, Lisa Veiber, Cedric Lothritz, Tegawendé F. Bissyandé, Jacques Klein, A comparison of pre-trained language models for multi-class text classification in the financial domain, *Companion Proceedings of the Web Conference*, 2021

# List of Published Models and Datasets

**Published Models:**

- LuxemBERT, the first Luxembourgish BERT model, available at `https://huggingface.co/lothritz/LuxemBERT` (2022)
- Lb_mBERT, a Luxembourgish BERT model derived from multilingual BERT, available at `https://huggingface.co/lothritz/Lb_mBERT`(2023)
- Lb_GottBERT, a Luxembourgish BERT model derived from GottBERT, available at `https://huggingface.co/lothritz/Lb_GottBERT` (2023)

**Published Datasets:**

- *Banking Client Support Dataset* for Intent Classification and Slot Filling, available at `https://github.com/Trustworthy-Software/BCS-dataset` (2021)
- *Luxembourgish POS dataset* for Part-of-Speech Tagging, available at `https://github.com/Trustworthy-Software/LuxemBERT` (2022)
- *Luxembourgish NER dataset* for Named Entity Recognition, available at `https://github.com/Trustworthy-Software/LuxemBERT` (2022)
- *Luxembourgish RTL News Classification dataset* for Text Classification, available at `https://github.com/Trustworthy-Software/LuxemBERT` (2022)
- *Luxembourgish WNLI*, translated from WNLI [61] , available at `https://github.com/Trustworthy-Software/LuxemBERT` (2022)
- *Luxembourgish Sentence Negation* for negating sentences, available at `https://github.com/Trustworthy-Software/LuxemBERT/tree/main` (2023)
- *Luxembourgish RTE*, translated from RTE [34], available at `https://github.com/Trustworthy-Software/LuxemBERT/tree/main` (2023)
- *Luxembourgish SST-2*, translated from SST-2 [26], available at `https://github.com/Trustworthy-Software/LuxemBERT/tree/main` (2023)
- *Luxembourgish Sentiment Analysis* for sentiment analysis, available at `https://github.com/Trustworthy-Software/LuxemBERT/tree/main` (2023)

# Bibliography

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 6000–6010, Curran Associates Inc., 2017.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *2019 Conference of the North American Chapter of the ACL: Human Language Technologies*, 2019.

[4] S. Vajjala, B. Majumder, A. Gupta, and H. Surana, *Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems.* O'Reilly Media, 2020.

[5] C. Döhmer, *Aspekte der luxemburgischen Syntax.* Melusina Press, 2020.

[6] S. Ehrhart and F. Fehlen, "Luxembourgish: A success story? a small national language in a multilingual country," *Handbook of language and ethnic identity*, pp. 285–298, 2011.

[7] K. Horner and J. J. Weber, "The language situation in luxembourg," *Current issues in language planning*, vol. 9, no. 1, pp. 69–128, 2008.

[8] P. Gilles, "Luxembourgish," in *The Oxford Encyclopedia of Germanic Linguistics*, 2022.

[9] P. Regulation, "Regulation (EU) 2016/679 of the European Parliament and of the Council," *Regulation (eu)*, vol. 679, p. 2016, 2016.

[10] A. S. Doğruöz, S. Sitaram, B. E. Bullock, and A. J. Toribio, "A survey of code-switching: Linguistic and social perspectives for language technologies," *arXiv preprint arXiv:2301.01967*, 2023.

[11] A. Herkenrath, "Receptive multilingualism in an immigrant constellation: Examples from turkish–german children's language," *International journal of bilingualism*, vol. 16, no. 3, pp. 287–314, 2012.

[12] P. Gilles, "Luxembourgish dialect classifications," *Dialectologia, Special Issue X: DIACLEU. An introduction to dialect classifications in Europe*, pp. 231–253, 2023.

[13] P. Gilles, “Mündlichkeit und schriftlichkeit in der luxemburgischen sprachgemeinschaft,” *Medien des Wissens. Interdisziplinäre Aspekte von Medialität*, pp. 43–64, 2011.

[14] C. Purschke, “Attitudes toward multilingualism in luxembourg. a comparative analysis of online news comments and crowdsourced questionnaire data,” *Frontiers in Artificial Intelligence*, vol. 3, p. 536086, 2020.

[15] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, “Syntactic Clustering of the Web,” *Computer networks and ISDN systems*, vol. 29, no. 8-13, pp. 1157–1166, 1997.

[16] H. P. Luhn, “A Statistical Approach to Mechanized Encoding and Searching of Literary Information,” *IBM Journal of research and development*, vol. 1, no. 4, pp. 309–317, 1957.

[17] K. S. Jones, “A Statistical Interpretation of Term Specificity and its Application in Retrieval,” *Journal of documentation*, 1972.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” *Advances in neural information processing systems*, vol. 26, 2013.

[19] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[20] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017.

[21] D. Jurafsky and J. Martin, “Neural Networks and Neural Language Models,” *Speech and Language Processing, 3rd Edition (Draft)*, pp. 123–142, 2019.

[22] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep Contextualized Word Representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[23] J. Howard and S. Ruder, “Universal Language Model Fine-Tuning for Text Classification,” *arXiv preprint arXiv:1801.06146*, 2018.

[24] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ Questions for Machine Comprehension of Text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016.

[26] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive Deep Models for Semantic Compositionality Over a

Sentiment Treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv preprint arXiv:1409.0473*, 2014.

[28] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," *http://openai-assets.s3.amazonaws.com/research-covers/language-unsupervised/language_understanding_paper.pdf*, 2018.

[29] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "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*, pp. 19–27, 2015.

[30] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "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*, pp. 353–355, 2018.

[31] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," in *Advances in neural information processing systems*, pp. 5754–5764, 2019.

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A Robustly Optimized BERT pretraining Approach," *arXiv preprint arXiv:1907.11692*, 2019.

[33] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "Ernie 2.0: A Continual Pre-Training Framework for Language Understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 8968–8975, 2020.

[34] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale ReAding Comprehension Dataset from Examinations," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, 2017.

[35] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, 2020.

[36] D. Araci, "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models," *arXiv preprint arXiv:1908.10063*, 2019.

[37] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2898–2904, 2020.

[38] P. J. Ortiz Suárez, B. Sagot, and L. Romary, "Asynchronous Pipelines for Processing Huge Corpora on Medium to Low Resource Infrastructures," in

*Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019* (P. Bański, A. Barbaresi, H. Biber, E. Breiteneder, S. Clematide, M. Kupietz, H. Lüngen, and C. Iliadi, eds.), (Mannheim), pp. 9 – 16, Leibniz-Institut für Deutsche Sprache, 2019.

[39] J. Abadji, P. Ortiz Suarez, L. Romary, and B. Sagot, "Towards a Cleaner Document-Oriented Multilingual Crawled Corpus," *arXiv e-prints*, p. arXiv:2201.06642, Jan. 2022.

[40] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah, and B. Sagot, "CamemBERT: a Tasty French Language Model," in *ACL 2020-58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[41] R. Scheible, F. Thomczyk, P. Tippmann, V. Jaravine, and M. Boeker, "GottBERT: a pure German Language Model," *arXiv preprint arXiv:2012.02110*, 2020.

[42] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez, "Spanish Pre-Trained BERT Model and Evaluation Data," in *PML4DC at ICLR 2020*, 2020.

[43] S. Wu and M. Dredze, "Are All Languages Created Equal in Multilingual BERT?," in *5th Workshop on Representation Learning for NLP*, pp. 120–130, 2020.

[44] G. Attardi, "WikiExtractor." `https://github.com/attardi/wikiextractor`, 2015.

[45] D. Goldhahn, T. Eckart, U. Quasthoff, *et al.*, "Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages.," in *LREC*, vol. 29, pp. 31–43, 2012.

[46] C. Cieri, M. Maxwell, S. Strassel, and J. Tracey, "Selection criteria for low resource language programs," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 4543–4549, 2016.

[47] A. S. Doğruöz and S. Sitaram, "Language technologies for low resource languages: Sociolinguistic and multilingual insights," in *Language Resources and Evaluation Conference (LREC)*, pp. 92–97, 2022.

[48] K. Bali, S. Sitaram, S. Cuendet, and I. Medhi, "A hindi speech recognizer for an agricultural video search application," in *Proceedings of the 3rd ACM Symposium on Computing for Development*, pp. 1–8, 2013.

[49] S. Kobayashi, "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)*, pp. 452–457, 2018.

[50] J. Wei and K. Zou, "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)*, pp. 6382–6388, 2019.

[51] R. Liu, G. Xu, C. Jia, W. Ma, L. Wang, and S. Vosoughi, “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)*, pp. 9031–9041, 2020.

[52] B. Santorini, “Part-of-Speech Tagging Guidelines for the Penn Treebank Project,” 1990.

[53] D. A. Dahl, M. Bates, M. K. Brown, W. M. Fisher, K. Hunicke-Smith, D. S. Pallett, C. Pao, A. Rudnicky, and E. Shriberg, “Expanding the Scope of the ATIS Task: The ATIS-3 Corpus,” in *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[54] R. Misra, “News Category Dataset,” *arXiv preprint arXiv:2209.11429*, 2022.

[55] R. Misra and J. Grover, *Sculpting Data for ML: The first act of Machine Learning.* 01 2021.

[56] E. S. Ristad, “Computational Structure of Human Language,” *Doctoral dissertation. Cambridge, Massachusetts*, 1990.

[57] A. Warstadt, A. Singh, and S. R. Bowman, “Neural Network Acceptability Judgments,” *arXiv preprint arXiv:1805.12471*, 2018.

[58] W. B. Dolan and C. Brockett, “Automatically Constructing a Corpus of Sentential Paraphrases,” in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[59] A. Williams, N. Nangia, and S. Bowman, “A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, Association for Computational Linguistics, 2018.

[60] R. B. Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor, “The Second Pascal Recognising Textual Entailment Challenge,” in *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, vol. 7, 2006.

[61] H. Levesque, E. Davis, and L. Morgenstern, “The Winograd Schema Challenge,” in *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

[62] A. Akbik, D. Blythe, and R. Vollgraf, “Contextual String Embeddings for Sequence Labeling,” in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1638–1649, 2018.

[63] M. Fleischman and E. Hovy, “Fine Grained Classification of Named Entities,” in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pp. 1–7, Association for Computational Linguistics, 2002.

[64] X. Ling and D. S. Weld, “Fine-Grained Entity Recognition,” in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI’12, p. 94–100, AAAI Press, 2012.

[65] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML ’01, (San Francisco, CA, USA), p. 282–289, Morgan Kaufmann Publishers Inc., 2001.

[66] K. Mai, T.-H. Pham, M. T. Nguyen, T. D. Nguyen, D. Bollegala, R. Sasano, and S. Sekine, “An Empirical Study on Fine-Grained Named Entity Recognition,” in *Proceedings of the 27th International Conference on Computational Linguistics*, (Santa Fe, New Mexico, USA), pp. 711–722, Association for Computational Linguistics, Aug. 2018.

[67] X. Ma and E. Hovy, “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, (Berlin, Germany), pp. 1064–1074, Association for Computational Linguistics, Aug. 2016.

[68] E. T. K. Sang and F. De Meulder, “Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition,” in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147, 2003.

[69] M. Peters, W. Ammar, C. Bhagavatula, and R. Power, “Semi-Supervised Sequence Tagging with Bidirectional Language Models,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1756–1765, 2017.

[70] H. B. Sahin, M. T. Eren, C. Tirkaz, O. Sonmez, and E. Yildiz, “English/Turkish Wikipedia Named-Entity Recognition and Text Categorization Dataset,” 2017.

[71] H. B. Sahin, C. Tirkaz, E. Yildiz, M. T. Eren, and O. Sonmez, “Automatically annotated Turkish corpus for Named Entity Recognition and Text Categorization using Large-scale Gazetteers,” *arXiv preprint arXiv:1702.02363*, 2017.

[72] N. Reimers and I. Gurevych, “Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Copenhagen, Denmark), pp. 338–348, 09 2017.

[73] N. Reimers and I. Gurevych, “Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks,” *arXiv preprint arXiv:1707.06799*, 2017.

[74] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, “HuggingFace’s Transformers: State-of-the-Art Natural Language Processing,” *ArXiv*, vol. abs/1910.03771, 2019.

[75] H. L. Guo, L. Zhang, and Z. Su, “Empirical Study on the Performance Stability of Named Entity Recognition Model across Domains,” in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 509–516, 2006.

[76] F. Béchet, A. Nasr, and F. Genet, “Tagging Unknown Proper Names Using Decision Trees,” in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 77–84, Association for Computational Linguistics, 2000.

[77] F. Bélanger and R. E. Crossler, “Privacy in the Digital Age: a Review of Information Privacy Research in Information Systems,” *MIS quarterly*, pp. 1017–1041, 2011.

[78] A. Act, “Health Insurance Portability and Accountability Act of 1996,” *Public law*, vol. 104, p. 191, 1996.

[79] J. C. Cuaresma, “The Gramm-Leach-Bliley act,” *Berkeley Tech. LJ*, vol. 17, p. 497, 2002.

[80] C. Song, T. Ristenpart, and V. Shmatikov, “Machine Learning Models that Remember too Much,” in *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pp. 587–601, 2017.

[81] S. M. Meystre, O. Ferrández, F. J. Friedlin, B. R. South, S. Shen, and M. H. Samore, “Text De-identification for Privacy Protection: a Study of its Impact on Clinical Text Information Content,” *Journal of biomedical informatics*, vol. 50, pp. 142–150, 2014.

[82] I. Tenney, J. Wexler, J. Bastings, T. Bolukbasi, A. Coenen, S. Gehrmann, E. Jiang, M. Pushkarna, C. Radebaugh, E. Reif, *et al.*, “The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 107–118, 2020.

[83] M. T. Ribeiro, S. Singh, and C. Guestrin, “" Why should I trust you?" Explaining the Predictions of any Classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[84] A. Bharadwaj, B. Ashar, P. Barbhaya, R. Bhatia, and Z. Shaikh, “Source Based Fake News Classification using Machine Learning,” 2020.

[85] D. Radev, “CLAIR Collection of Fraud Email (Repository)-ACL Wiki,” 2008.

[86] H. Berg, A. Henriksson, and H. Dalianis, “The Impact of De-identification on Downstream Named Entity Recognition in Clinical Text,” in *11th International Workshop on Health Text Mining and Information Analysis*, pp. 1–11, Association for Computational Linguistics, 2020.

[87] L. Deleger, K. Molnar, G. Savova, F. Xia, T. Lingren, Q. Li, K. Marsolo, A. Jegga, M. Kaiser, L. Stoutenborough, *et al.*, “Large-Scale Evaluation of Automated Clinical Note De-identification and its Impact on Information

Extraction," *Journal of the American Medical Informatics Association*, vol. 20, no. 1, pp. 84–94, 2013.

[88] X. Ouyang, S. Wang, C. Pang, Y. Sun, H. Tian, H. Wu, and H. Wang, "ERNIE-M: Enhanced Multilingual Representation by Aligning Cross-Lingual Semantics with Monolingual Corpora," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, (Online and Punta Cana, Dominican Republic), pp. 27–38, Association for Computational Linguistics, Nov. 2021.

[89] A. D. Taylor and A. M. Pacelli, *Mathematics and Politics: Strategy, Voting, Power, and Proof.* Springer Science & Business Media, 2008.

[90] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[91] D. Slijepčević, M. Henzl, L. D. Klausner, T. Dam, P. Kieseberg, and M. Zeppelzauer, "k-Anonymity in Practice: How Generalisation and Suppression affect Machine Learning Classifiers," *Computers & Security*, vol. 111, p. 102488, 2021.

[92] P. Samarati, "Protecting Respondents Identities in Microdata Release," *IEEE transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[93] R. Kohavi *et al.*, "Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-tree Hybrid.," in *Kdd*, vol. 96, pp. 202–207, 1996.

[94] J. S. Obeid, P. M. Heider, E. R. Weeda, A. J. Matuskowitz, C. M. Carr, K. Gagnon, T. Crawford, and S. M. Meystre, "Impact of De-identification on Clinical Text Classification using Traditional and Deep Learning Classifiers," *Studies in health technology and informatics*, vol. 264, p. 283, 2019.

[95] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open Source Language Understanding and Dialogue Management," *arXiv preprint arXiv:1712.05181*, 2017.

[96] W. Xu, B. Haider, and S. Mansour, "End-to-End Slot Alignment and Recognition for Cross-Lingual NLU," in *Proceedings of EMNLP 2020*, (Online), pp. 5052–5063, ACL, Nov. 2020.

[97] S. Upadhyay, M. Faruqui, G. Tür, H. Dilek, and L. Heck, "(Almost) Zero-Shot Cross-Lingual Spoken Language Understanding," in *2018 IEEE ICASSP*, pp. 6034–6038, 2018.

[98] D. Chicco and G. Jurman, "The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation," *BMC genomics*, vol. 21, pp. 1–13, 2020.

[99] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal Classifier for Imbalanced Data using Matthews Correlation Coefficient Metric," *PloS one*, vol. 12, no. 6, p. e0177678, 2017.

[100] X. Wan, "Co-training for Cross-Lingual Sentiment Classification," in *Joint Conference of the 47th Annual Meeting of the ACL*, pp. 235–243, 2009.

[101] S. Lauly, H. Larochelle, M. M. Khapra, B. Ravindran, V. Raykar, A. Saha, *et al.*, "An Autoencoder Approach to Learning Bilingual Word Representations," *arXiv preprint arXiv:1402.1454*, 2014.

[102] G. Zhou, T. He, and J. Zhao, "Bridging the Language Gap: Learning Distributed Semantics for Cross-Lingual Sentiment Classification," in *International Conference on Natural Language Processing and Chinese Computing*, pp. 138–149, Springer, 2014.

[103] H. Zhou, L. Chen, F. Shi, and D. Huang, "Learning Bilingual Sentiment Word Embeddings for Cross-Language Sentiment Classification," in *53rd Annual Meeting of the ACL and the 7th International Joint Conference on NLP*, pp. 430–440, 2015.

[104] C. Abbet, M. M'hamdi, A. Giannakopoulos, R. West, A. Hossmann, M. Baeriswyl, and C. Musat, "Churn Intent Detection in Multilingual Chatbot Conversations and Social Media," *arXiv preprint arXiv:1808.08432*, 2018.

[105] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual Transfer Learning for Multilingual Task Oriented Dialog," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3795–3805, 2019.

[106] M. Vanjani, J. Posey, and M. Aiken, "An Evaluation of a Multilingual Chatbot," *Issues in Information Systems*, vol. 20, no. 1, pp. 134–143, 2019.

[107] M. Vanjani, M. Aiken, and M. Park, "Chatbots for multilingual conversations," *Journal of Management Science and Business Intelligence*, vol. 4, no. 1, pp. 19–24, 2019.

[108] Z. Lin, Z. Liu, G. I. Winata, S. Cahyawijaya, A. Madotto, Y. Bang, E. Ishii, and P. Fung, "XPersona: Evaluating Multilingual Personalized Chatbot," *arXiv preprint arXiv:2003.07568*, 2020.

[109] S. Upadhyay, M. Faruqui, G. Tür, H.-T. Dilek, and L. Heck, "(almost) Zero-Shot Cross-Lingual Spoken Language Understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6034–6038, IEEE, 2018.

[110] C. Costello, R. Lin, V. Mruthyunjaya, B. Bolla, and C. Jankowski, "Multi-Layer Ensembling Techniques for Multilingual Intent Classification," *arXiv preprint arXiv:1806.07914*, 2018.

[111] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a Distilled Version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[112] D. Bernhard and A.-L. Ligozat, "Hassle-Free POS-Tagging for the Alsatian Dialects," 2013.

[113] A. Carnie, *Syntax: A Generative Introduction.* John Wiley & Sons, 2021.

[114] J. W. Pennebaker, “The Secret Life of Pronouns,” *New Scientist*, vol. 211, no. 2828, pp. 42–45, 2011.

[115] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos, “Management of an Academic HPC Cluster: The UL Experience,” in *Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014)*, (Bologna, Italy), pp. 959–967, IEEE, July 2014.

[116] P. Gilles, “Phonologie der n-Tilgung im Moselfränkischen (’Eifler Regel’). Ein Beitrag zur dialektologischen Prosodieforschung,” *Perspektiven einer linguistischen Luxemburgistik. Studien zur Diachronie und Synchronie*, pp. 29–68, 2006.

[117] J. Demšar, “Statistical Comparisons of Classifiers over Multiple Data Sets,” *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[118] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences.* Academic press, 2013.

[119] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised Cross-Lingual Representation Learning at Scale,” *CoRR*, vol. abs/1911.02116, 2019.

[120] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, “A Survey on Recent Approaches for Natural Language Processing in Low-resource Scenarios,” *arXiv preprint arXiv:2010.12309*, 2020.

[121] B. Muller, A. Anastasopoulos, B. Sagot, and D. Seddah, “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*, pp. 448–462, 2021.

[122] I. Olariu, “Evaluating Data Augmentation Techniques for Improving the Training of Low-reource Language Models [Unpublished Master’s Thesis],” University of London, 2022.

[123] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, “Beyond Accuracy: Behavioral Testing of NLP Models with CheckList,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, 2020.

[124] G. N. C. Tejada, J. Scholtes, and G. Spanakis, “A Study of BERT’s Processing of Negations to Determine Sentiment,” in *33rd Benelux Conference on Artificial Intelligence and 30th Belgian-Dutch Conference on Machine Learning*, pp. 47–59, 2021.

[125] S. Schweter, “BERTurk - BERT Models for Turkish,” Apr. 2020.

[126] M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, *et al.*, “No Language Left Behind: Scaling Human-Centered Machine Translation,” *arXiv preprint arXiv:2207.04672*, 2022.