

A Data-driven Approach to Natural Language Processing for Contemporary and Historical French

Pedro Ortiz Suarez

▶ **To cite this version:**

Pedro Ortiz Suarez. A Data-driven Approach to Natural Language Processing for Contemporary and Historical French. Document and Text Processing. Sorbonne Université, 2022. English. NNT : 2022SORUS155 . tel-03770337v2

HAL Id: tel-03770337
https://theses.hal.science/tel-03770337v2

Submitted on 7 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.





Sorbonne Université

Ecole Doctorale Informatique, Télécommunications et Electronique - ED130

Inria de Paris / Équipe ALMAnaCH

Thèse de doctorat
Discipline : Informatique

Présentée par

Pedro Ortiz Suarez

Dirigée par

Laurent Romary et **Benoît Sagot**

Pour obtenir le grade universitaire de

Docteur de Sorbonne Université

A Data-driven Approach to Natural Language Processing for Contemporary and Historical French

---

Présentée et soutenue publiquement le 27 juin 2022 devant le jury composé de :

Francis Bach	Inria - SIERRA	Examinateur
Maud Ehrmann	EPFL	Examinateur
Alexander Geyken	BBAW	Examinateur
Anna Korhonen	University of Cambridge	Rapporteur
Laurent Romary	Inria - ALMAnaCH	Directeur
Benoît Sagot	Inria - ALMAnaCH	Directeur
Holger Schwenk	Meta AI Research	Rapporteur

Une approche basée sur les données pour le traitement automatique du langage naturel en français contemporain et historique

Résumé

Depuis plusieurs années, les approches neuronales ont régulièrement amélioré l'état de l'art du traitement automatique des langues (TAL) sur une grande variété de tâches. L'un des principaux facteurs ayant permis ces progrès continus est l'utilisation de techniques d'apprentissage par transfert. Ces méthodes consistent à partir d'un modèle pré-entraîné et à le réutiliser, avec peu ou pas d'entraînement supplémentaire, pour traiter d'autres tâches. Même si ces modèles présentent des avantages évidents, leur principal inconvénient est la quantité de données nécessaire pour les pré-entraîner. Ainsi, le manque de données disponibles à grande échelle a freiné le développement de tels modèles pour le français contemporain et a fortiori pour ses états de langue plus anciens.

Cette thèse met l'accent sur le développement de corpus pour le pré-entraînement de telles architectures. Cette approche s'avère extrêmement efficace car nous sommes en mesure d'améliorer l'état de l'art pour un large éventail de tâches de TAL pour le français contemporain et historique, ainsi que pour six autres langues contemporaines. De plus, nous montrons que ces modèles sont extrêmement sensibles à la qualité, à l'hétérogénéité et à l'équilibre des données de pré-entraînement et montrons que ces trois caractéristiques sont de meilleurs prédicteurs de la performance des modèles que la taille des données de pré-entraînement. Nous montrons également que l'importance de la taille des données de pré-entraînement a été surestimée en démontrant à plusieurs reprises que l'on peut pré-entraîner de tels modèles avec des corpus de taille assez modeste.

Mots-clés : modèle de langue, corpus de pré-entraînement, traitement automatique des langues, français contemporain, français historique, apprentissage par transfert.

A Data-driven Approach to Natural Language Processing for Contemporary and Historical French

Abstract

In recent years, neural methods for Natural Language Processing (NLP) have consistently and repeatedly improved the state of the art in a wide variety of NLP tasks. One of the main contributing reasons for this steady improvement is the increased use of transfer learning techniques. These methods consist in taking a pre-trained model and reusing it, with little to no further training, to solve other tasks. Even though these models have clear advantages, their main drawback is the amount of data that is needed to pre-train them. The lack of availability of large-scale data previously hindered the development of such models for contemporary French, and even more so for its historical states.

In this thesis, we focus on developing corpora for the pre-training of these transfer learning architectures. This approach proves to be extremely effective, as we are able to establish a new state of the art for a wide range of tasks in NLP for contemporary, medieval and early modern French as well as for six other contemporary languages. Furthermore, we are able to determine, not only that these models are extremely sensitive to pre-training data quality, heterogeneity and balance, but we also show that these three features are better predictors of the pre-trained models' performance in downstream tasks than the pre-training data size itself. In fact, we determine that the importance of the pre-training dataset size was largely overestimated, as we are able to repeatedly show that such models can be pre-trained with corpora of a modest size.

Keywords: language model, pre-training corpora, natural language processing, contemporary french, historical french, transfer learning.

Acknowledgments

First and foremost I am extremely grateful to my supervisors Laurent Romary and Benoît Sagot, who at the beginning of my Ph.D. accepted to supervise me even though I was coming from a different background and had absolutely no experience in Natural Language Processing (NLP) or Digital Humanities (DH). You both had the patience to teach me and accompany me throughout this Ph.D. journey, supporting me not only academically but also personally; you're both my mentors and for that I am extremely grateful. To Benoît, thank you so much for always being there during all the difficult deadlines ready to correct and advice. To Laurent, thank you so much for all your invaluable advise which undeniably helped go through the difficult situations that I encountered during this Ph.D.

I would like to thank all the members of the committee for agreeing to evaluate this thesis and its defense, for their comments, advice and questions which have greatly contributed to the improvement and finalization of this work. Special thanks to Anna Korhonen and Holger Schwenk, for accepting to read my manuscript in such a short span of time and for the helpful comments in your pre-reports. I would also like to thank the members of my *comité de suivi de thèse*, for their support and their advice for the end of the thesis.

I would also like to thank the ANR BASNUM project (ANR-18-CE38-0003) for founding my Ph.D. thesis, as well as its members Geoffrey Williams, Ioana Galleron, Mohamed Khemakhem and Clarissa Stincone, with whom I had the pleasure of collaborating at the beginning of this Ph.D. thesis.

This thesis is not only the product of my work, but the fruit of numerous collaborations without which it would have never been possible. For my collaborators and coauthors, I would like to thank all the members of the *CamemBERT* project, specially to Louis and Benjamin. We managed to publish a valuable and influential resource for French NLP and even for NLP in languages other than English. I will always be grateful to have participated in this project.

I also want to mention Rachel, Philippe and Alexandre from the *Early Modern French* project; Tian Tian and Gaël Lejeune from the *CLEF-HIPE2020* shared task; and Benoît Crabbé from the *BERTrade* project. You taught me many things during our time collaborating together, and I cannot say enough how lucky I was to have the opportunity to work with you and how grateful I am to all of you.

To Simon Gabay, with whom I collaborated during the last part of my Ph.D. thesis, I'd like to thank for providing me with so much high quality data. Most of the experiments at the end of this manuscript would have never been possible without your work, so for this you have my whole gratitude.

To all the people at Masakhane and all the coauthors of the *Quality at Glance* paper, specially to Julia and Isaac; I truly have no words to express how lucky I was to work with such an amazing group of people, you taught me about the importance of

looking at my own data, and you also helped me restart the OSCAR project. Thank you so much for working with me and expressing interest in my corpus.

As the OSCAR project became larger, I no longer had the means to continue working on it alone, this is why I would like to express my deepest gratitude to Julien. Thank you so much for your patience, for taking the time to go through my messy code and making it readable, for managing our users while I was away and for taking care of OSCAR like it was your own. I only have gratitude and respect for you, you're truly one of the greatest engineers that I know, and even though we have sadly never met in person, know that you became a valuable friend and colleague to me during this last year.

To all my other colleagues at the ALMAnaCH team that I have worked with such as Djamé, Eric, Syrielle, Kim and Meriem; and to the newcomers such as Hugo, Rua and Roman. Thank you for making of Inria such a nice place to work at.

At the beginning of this Ph.D. thesis I was just a mathematician that did not have any clue about NLP or Machine Learning. However, I encountered a group of people that believed in me and supported me in working at Inria and starting this Ph.D. thesis in NLP. For this I would like to thank in particular Patrice, Alba and Professor Isis Truck.

While this Ph.D. thesis was an academic exercise, I cannot overstate how valuable was the personal support that the friends I made here in Paris gave me during the last four years. You made me feel at home in France and you actually helped me become one of your own; you accompanied me during very turbulent personal times and even through the pandemic. Some of you even became coauthors and collaborators. Thank you so much Yoann and Ariane, Alix and Jen, Clementine, Loïc, Murielle, Manon, Emilia, Ganesh, Tanti and Arij. I do not have words to truly express to you how thankful I am.

To all my Colombian friends that have always supported me, and specially to Zubieta, Juan Rafael, Pedro Pablo, Isabel, Susana, Henao, Pablo, Andrés, Maritza Carlos and Min. Thank you so much for sending me all your support from afar.

To my family, specially to my father, my grandmother, to Elena and Gloria and to my little Family in the US, Fercho and Beatriz. Thank you so much for all your support and your love, in spite of the distance and the pandemic keeping us apart for years, you were always there when I needed it the most. Thank you so much for always believing in me.

Finally, I'd like to thank Mathilde and her family, Paul, Anne, Thierry and Anne-Marie. Since the beginning you always welcomed me with open arms, and you have supported and accompanied me during the last two years. I have no words to express how grateful I am for everything that you have done for me and for the hospitality that you have always offered me. To Mathilde, you have all my gratitude for your constant support and specially for putting up with me during the last days

of this Ph.D. thesis, I know it was difficult, but I do believe good times are ahead of us.

Contents

Résumé . . . iii
Abstract . . . iv
Acknowledgments . . . v

I Introduction and Related Work 1

1 Introduction 3
1.1 The BASNUM Project . . . 3
1.2 Transfer Learning in NLP . . . 5
1.3 Digital Humanities and NLP for Historical Languages . . . 7
1.3.1 Medieval French . . . 8
1.3.2 Early Modern French . . . 10
1.4 Outline . . . 11

2 On Raw Corpora for Language Modeling 13
2.1 Common Crawl . . . 13
2.1.1 FastText's Pipeline . . . 14
2.1.2 CCNet . . . 15
2.2 Demand for Large Corpora . . . 16
2.2.1 Problems with Crawled Corpora . . . 17
2.2.2 Publically Available Web-based Corpora . . . 17
2.3 Historical French Corpora . . . 19

3 On Language Models and Downstream Tasks 21
3.1 On Word Representations . . . 21
3.1.1 Some Language Model Architectures . . . 23
3.2 Non-English Word Representations . . . 24
3.2.1 Recent Developments in Contextualized Representations . . . 25
3.3 Downstream Tasks Evaluation . . . 26
3.3.1 Evaluation Datasets . . . 28
3.3.2 Brief State of the Art for Dependency Parsing and POS Tagging 30
3.3.3 Brief State of the Art for NER . . . 30
3.3.4 Common Baselines . . . 31
3.4 Neural Models for Historical Languages . . . 32

II OSCAR 33

4 Goclassy: an Asynchronous Language Classification Pipeline for Common Crawl 35
4.1 An Asynchronous Pipeline . . . 35
4.2 Benchmarks . . . 37
4.3 OSCAR 2019 . . . 38
4.4 Conclusions . . . 39

5 A First Evaluation of the OSCAR Corpus 41
5.1 Corpora . . . 41
5.1.1 Noisiness . . . 42
5.2 Experimental Setting . . . 43
5.2.1 Contextualized word embeddings . . . 44
5.2.2 UDPipe 2.0 . . . 44
5.2.3 Treebanks . . . 45
5.3 Results & Discussion . . . 46
5.3.1 Parsing and POS tagging results . . . 46
5.3.2 Impact of the number of training epochs . . . 47
5.4 Conclusions . . . 49

6 Quality at a Glance: An Audit of OSCAR 2019 and other Web-Crawled Datasets 51
6.1 Auditing Data Quality . . . 52
6.1.1 Auditing Process . . . 52
6.1.2 Human Audit Results . . . 54
6.1.3 Automatic Filtering . . . 59
6.2 Dataset Mis-labeling . . . 59
6.3 Risks of Low-Quality Data . . . 61
6.4 Future Work and Recommendations . . . 62
6.5 Conclusions for the OSCAR Project . . . 63

7 Ungoliant: The Second OSCAR pipeline 65
7.1 Limitations of the OSCAR 2019 Corpus and its Generation Pipeline . 66
7.1.1 OSCAR 2019 . . . 66
7.1.2 goclassy . . . 66
7.2 Building a New Version of the OSCAR Corpus . . . 67
7.2.1 Ungoliant . . . 67
7.2.2 Iterating on the goclassy Pipeline . . . 68
7.2.3 Characteristics of the OSCAR 21.09 Corpus . . . 71
7.2.4 License . . . 74
7.3 Conclusion . . . 75

8 Towards a Cleaner Document-Oriented Annotated OSCAR Corpus 77
8.1 Filtering . . . 78
8.1.1 Header and footer filter . . . 78
8.1.2 Short lines proportion filter . . . 79
8.2 Identification . . . 79
8.2.1 Multilingual document identification . . . 79
8.2.2 Monolingual identification . . . 80
8.3 Annotation . . . 81
8.3.1 Length-based annotations . . . 81
8.3.2 Noise detection . . . 81
8.3.3 Adult documents . . . 82
8.4 Corpus . . . 82
8.4.1 Comparison with OSCAR 21.09 . . . 83
8.4.2 Annotations . . . 86
8.5 Discussion . . . 90
8.5.1 Corpus . . . 90
8.5.2 Annotations . . . 91
8.6 Conclusion . . . 91

III French Corpora 93

9 Contemporary French Corpora 95
9.1 Contemporary French Balanced Corpora . . . 96
9.1.1 CaBeRnet . . . 96
9.1.2 French Children Book Test (CBT-fr) . . . 98
9.1.3 Descriptive Comparison . . . 99
9.2 A named entity annotation layer for the UD version of the French TreeBank . . . 101
9.2.1 Alignment to the UD version of the FTB . . . 102
9.3 Conclusion . . . 103

10 Historical French Corpora 105
10.1 Medieval French Corpus . . . 106
10.2 Early Modern French Corpora . . . 107
10.2.1 FreEM$_{max}$ . . . 107
10.2.2 FreEM$_{LPM}$ . . . 111
10.2.3 FreEM$_{NER}$ . . . 111
10.3 Conclusion . . . 114

IV Models and Evaluation 115

11 CamemBERT 117
11.1 CamemBERT: A Contemporary French Language Model 118
11.1.1 Training data 118
11.1.2 Pre-processing 118
11.1.3 Language Modeling 118
11.1.4 Using CamemBERT for downstream tasks 120
11.2 Evaluation of CamemBERT 121
11.3 Impact of corpus origin and size 124
11.3.1 Common Crawl vs. Wikipedia? 125
11.3.2 How much data do you need? 126
11.4 Discussion 127
11.5 Conclusion 127

12 FrELMo 129
12.1 FrELMo 130
12.1.1 Benchmarking NER Models 130
12.1.2 Results 132
12.1.3 Impact of shuffling the data 134
12.1.4 Conclusions of the Benchmark 135
12.2 Pre-training Corpora Evaluation for ELMo models 135
12.2.1 ELMo Pre-traing & Fine-tuning Method 136
12.2.2 Results & Discussion 136
12.2.3 Conclusion 139

13 SinNer CLEF-HIPE2020 141
13.1 Dataset for the CLEF-HIPE shared task 141
13.2 CRFs and Contextualized Word Embeddings for NER 143
13.2.1 CRF model (run3) 143
13.2.2 ELMo-LSTM-CRF (run1 and run2) 144
13.3 Results and Discussion 145
13.3.1 Official shared task results 145
13.3.2 Study of sequence segmentation 145
13.3.3 To dev or not to dev? 146
13.4 Conclusion 147

14 BERTrade 149
14.1 Experiments 150
14.1.1 Baselines 151
14.1.2 With related contextual embeddings 152
14.1.3 With raw linguistic data 152

14.1.4 Putting it all together . . . 153
14.2 Conclusion . . . 154

15 D'AlemBERT 155
15.1 D'AlemBERT: a neural language model for Early Modern French . . . 155
15.1.1 Pre-processing . . . 156
15.1.2 Language Modelling . . . 156
15.2 Evaluation and Discussion . . . 157
15.2.1 Part-Of-Speech Tagging . . . 157
15.2.2 Named Entity Recognition . . . 158
15.3 Conclusion . . . 159

V Conclusions and Perspectives 161

16 Conclusions and Perspectives 163

VI Appendices 165

A Goclassy: an Asynchronous Language Classification Pipeline for Common Crawl 167

B A First Evaluation of the OSCAR Corpus 171
B.1 Computational cost and carbon footprint . . . 171
B.2 Number of training steps for each checkpoint and each corpus . . . 173

C Quality at a Glance: An Audit of OSCAR 2019 and other Web-Crawled Datasets 175
C.1 Details on Language Code Issues . . . 175
C.2 Complete Error Taxonomy and Instructions . . . 178
C.3 Methodological Notes . . . 179
C.4 Complete Audit Results . . . 180

D Towards a Cleaner Document-Oriented Annotated OSCAR Corpus 183
D.1 Carbon Footprint . . . 183
D.2 Language Table . . . 184

E CamemBERT 187
E.1 Impact of Whole-Word Masking . . . 187
E.2 Impact of model size . . . 187
E.3 Impact of training dataset . . . 188
E.4 Impact of number of steps . . . 188

F BERTrade 191
F.1 Collecting the Data . . . 191
F.2 Details on the Models . . . 192
F.2.1 Models Trained From Scratch . . . 192
F.2.2 Post-training . . . 192
F.3 Carbon Footprint . . . 192

G D'AlemBERT 195
G.1 Carbon Footprint . . . 195
G.2 Detail Results of Experiments in NER by Entity Type . . . 196
G.3 Entity Distribution by Text in NER Data . . . 196

Bibliography 201

License 242

Back 243

To both my grandmother and my mother.

Part I

Introduction and Related Work

1 Introduction

In which we present the BASNUM project and the motivations behind this Ph.D thesis. We give a small overview of our approach by discussing *transfer learning* in NLP, we then discuss Digital Humanities and NLP for historical languages, and we give a brief overview of the two historical states of the French language that will have a central part later in this Ph.D. thesis. Finally, we give the outline of the present text.

1.1 The BASNUM Project

This thesis is part of the ANR BASNUM project (ANR-18-CE38-0003), which had as its main objective to digitize the "*Dictionnaire Universel*" (DU) of Antoine Furetière, in its 1701 version reviewed and corrected by Basnage de Beauval (Furetière, 1701), and to analyze it with digital tools, in order to reveal the importance of this work for the evolution of science and mentalities in the 18th century. The project also aimed to contribute to the current movement to design innovative methods for digitizing, encoding and analyzing texts.

From a purely computational point of view, the BASNUM project intended to carry out two types of tasks:

1. a first structuring task where the macrostructure of the dictionary would be annotated,

2. a second enrichment task which consisted in carrying out a wide range of tasks of information extraction, annotation of the dictionary microstructure and even normalization and modernization of the text.

The first task of automatically structuring dictionaries had been already partially covered by the work of Khemakhem et al. (2017, 2018) who developed *GROBID-dictionaries*, a submodule of *GROBID*[1] (Grobid contributors, 2008 — 2018) implementing a Java machine learning library for structuring digitized lexical resources in *TEI* format (TEI Consortium, eds, 2018), to enable analysis, extraction and structuring of textual information in such resources. *GROBID-dictionaries* had already

[1]Machine learning library to extract, analyze and restructure raw documents such as PDFs into structured and TEI-encoded documents.

obtained promising results and performances (Khemakhem, 2020), so much so that it was used to make a first annotation of the *Dictionnaire Universel* macrostructure.

Given the work done by Khemakhem (2020), we decided to concentrate on the second task of enriching the dictionaries, which at the time remained quite general and abstract, in particular in contrast to the first task of structuring. To approach this task, we had two options: either developing multiple models and annotation systems dedicated to each of the subtasks involved in this *enrichment* and solely targeting the *Dictionnaire Universel*, or developing a single generic annotation model capable of addressing all enrichment subtasks and capable of handling not only the *Dictionnaire Universel*, but also other texts and resources from the modern period.[2]

Given the nature of the enrichment task and the fact that new neural language models able to transfer knowledge between different tasks in natural language processing (NLP) had just been published at the beginning of this thesis (Peters et al., 2018; Devlin et al., 2019), we decided to focus on the second option and, accordingly, to develop a single general model that we hoped would be able to be used for all these enrichment tasks and for any type of document in Modern or Contemporary French.

By choosing this approach, we also wanted to approach in an indirect way the first task of automatic structuring. This is because we believed that it was possible to improve the first results of *GROBID*-dictionaries by using new neural models. Indeed, *GROBID*-dictionaries relied on CRF models (Conditional Random Fields) (Lafferty et al., 2001) which were widely used for token labeling and classification, but that had been superseded by these neural models in recent years (Lample et al., 2016; Devlin et al., 2019). Furthermore, we knew that the developers of *GROBID* had started working with some of these neural models by writing *DeLFT*, a library for text processing, covering token labeling and classification. This library reimplements the latest machine learning models in NLP (DeLFT contributors, 2018) and aims to improve *GROBID*'s pipelines. It is tools and ideas like those contained in *DeLFT* that could be applied to *GROBID*-Dictionaries to significantly improve and expand its capabilities for the benefit of the *BASNUM* project, especially in addition to resources that we had decided to develop.

Having chosen to develop these new models for French, such as *ELMo* (Peters et al., 2018) or *BERT* (Devlin et al., 2019), we had to start by building and collecting our own corpus for the pre-training of these architectures, since the contemporary French corpora freely available at the time, such as Wikipedia or frWAC (Baroni et al., 2009), were not considered to be large enough for this (Liu et al., 2019).

Our plan then was to develop a pre-training corpus for Contemporary French, then to pre-train a language model for Contemporary French and finally to use the knowledge transfer capabilities of these architectures to adapt it to Early Modern French, in case we were unable to find enough textual resources to directly pre-

[2]Between the 16th and 18th century.

train such a language model for Early Modern French. During this thesis, we also wanted to investigate the question of the minimum amount of resources required to successfully pre-train such models, an amount which, at the time, was considered higher than what was available for historical languages (Peters et al., 2018; Liu et al., 2019).

1.2 Transfer Learning in NLP

In recent years neural methods for Natural Language Processing (NLP) have consistently and repeatedly improved the state of the art in a wide variety of NLP tasks such as parsing, PoS-tagging, named entity recognition, machine translation, text classification and reading comprehension among others. Probably the main contributing factor in this steady improvement for NLP models is the raise in usage of *transfer learning* techniques in the field. These methods normally consist of taking a pre-trained model and reusing it, with little to no retraining, to solve a different task from the original one it was intended to solve; in other words, one *transfers* the *knowledge* from one task to another.

Most of the transfer learning done in NLP nowadays is done in an unsupervised manner, that is, it normally consists of a *language model* that is fed unannotated plain text in a particular language; so that it *extracts* or *learns* the basic *features* and patterns of the given language. The model is subsequently used on top of an specialised architecture designed to tackle a particular NLP task. Probably the best known example of this type of model are *word embeddings* which consist of real-valued vector representations that are trained for each word on a given corpus. Some notorious examples of word embeddings are word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and fastText (Mikolov et al., 2018). All these models are *context-free*, meaning that a given word has one single vector representation that is independent of context, thus for a polysemous word like Washington, one would have one single representation that is reused for the city, the state and the US president.

In order to overcome the problem of polysemy, *contextual* models have recently appeared. Most notably ELMo (Peters et al., 2018) which produces deep contextualized word representations out of the internal states of a deep bidirectional language model in order to model word use and how the usage varies across linguistic contexts. ELMo still needs to be used alongside a specialised architecture for each given downstream task, but newer architectures that can be fine-tuned have also appeared. For these, the model is first fed unannotated data, and is then fine-tuned with annotated data to a particular downstream task without relying on any other architecture. some remarkable examples of this type of model are GPT-1, GPT-2 (Radford et al., 2018, 2019), BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019); the latter being the current state-of-the-art for multiple downstream tasks. All of these models are

different arrangements of the Transformer architecture (Vaswani et al., 2017) trained with different datasets, except for XLNet which is an instance of the Transformer-XL (Dai et al., 2019).

Even though these models have clear advantages, their main drawback is the amount of data that is needed to train them in order to obtain a functional and efficient model. For instance, for the first English version of word2vec, Mikolov et al. (2013) used a one billion word dataset consisting of various news articles. Later Al-Rfou' et al. (2013) and then Bojanowski et al. (2017) used the plain text from Wikipedia to train distributions of word2vec and fastText respectively, for languages other than English. Now, the problem of obtaining large quantities of data aggravates even more for contextual models, as they normally need multiple instances of a given word in order to capture all its different uses and in order to avoid overfitting due to the large quantity of hyperparameters that these models have. Peters et al. (2018) for example use a 5.5 billion token[3] dataset comprised of crawled news articles plus the English Wikipedia in order to train ELMo, Devlin et al. (2019) use a 3.3 billion word[4] corpus made by merging the English Wikipedia with the BooksCorpus (Zhu et al., 2015), and Radford et al. (2019) use a 40GB English corpus created by scraping outbound links from Reddit.[5]

While Wikipedia is freely available, and multiple pipelines exist[6,7] to extract plain text from it, some of the bigger corpora mentioned above are not made available by the authors either due to copyright issues or probably because of the infrastructure needed to serve and distribute such big corpora. Moreover, the vast majority of both these models and the corpora they are trained with are in English, meaning that the availability of high quality NLP for other languages, specially for low-resource and historical languages, is rather limited.

The problem of scarcity of pre-training and training data is something that we will have to weather throughout the course of this Ph.D. thesis, we will thus focus on developing these resources in order to try to fill the existing gap that we observed at the beginning of this thesis for both Contemporary and Historical French. In fact rather than focusing on improving the architectures of the available models mentioned above, we will focus on developing resources to both pre-train and fine-tune them, we will thus try to improve the state of the art for both Contemporary and Historical French by solely working on data and textual resources. We will study the impact that balanced corpora and corpus quality has on these models, we will try to answer the question of how much data is even needed in order to obtain a decently performing contextualized model, and we will also try to assess the impact

[3] Punctuation marks are counted as tokens.
[4] Space sparated tokens.
[5] https://www.reddit.com/
[6] https://github.com/attardi/wikiextractor
[7] https://github.com/hghodrati/wikifil

of fine-tuning (training) data on the performance of these models in a wide range of downstream tasks.

1.3 Digital Humanities and NLP for Historical Languages

With the rise of digital humanities, it is becoming increasingly important to develop high quality tools to automatically process old states of languages. Libraries, archives and museums, among others, are digitizing large numbers of historical sources, from which high quality data must be extracted for further study by specialists of human sciences following new approaches such as "distant reading" (Moretti, 2013). Many (sub)tasks such as automatic OCR post-correction (Rijhwani et al., 2021) and linguistic annotation (Camps et al., 2021) benefit from pre-trained language models to improve their accuracy.

Languages evolve over time on many levels: from one century to another, we see variations in spelling, syntax, the lexicon etc. However, this variation is not uniform: it tends, at least for "literate scriptors" (literature, journalism, law, etc.), to converge towards a single norm over time, and this has especially been the case for French because of the prominent role of the *Académie française* and the *remarqueurs* (Ayres-Bennett and Seijido, 2011). The result of this convergence is, for instance, that spelling and word order within sentences have become stricter, where they were less so in the past. From a computational perspective, historical states of language are therefore not only different from the contemporary state, but, from a computational perspective, are also more complex because they do not follow a strict and explicit norm. In French, this explicit norm appeared in the 17th c. and was slowly integrated throughout the 18th c.

On top of this first linguistic problem, a second issue appears: because the production of textual sources has continued to grow exponentially, it is easier to collect a corpus for contemporary French than for the 19th c. French, which is itself easier than for the 18th c. French, etc. The further we go back in time, the more scarce resources are, which creates the following paradox: we have more data when the language is homogeneous and simple for the computer to process, and less when it is heterogeneous and harder to process.

Using contextual word embeddings as input representations has brought clear gains in performances for most of the NLP tasks for which they have been used. However, this has mostly been attested in languages where sufficient (raw) linguistic data is available. For less-resourced languages, the most common approach has been to leverage multilingual models such as mBERT (Devlin et al., 2019) whose pre-training corpus does not even contain historical data.

Historical languages are typical cases where available linguistic data is limited, with no chance of acquiring new texts. They are also not normalized by spelling and institutional conventions and tend to be more heterogeneous than contemporary

lesser-resourced languages, giving us a very interesting yet challenging case of study to try to approach with contextualized word embedding models.

1.3.1 Medieval French

Medieval French covers both Old French (9th-13th c.) and Middle French (14th-15th c.). These stages are linguistically close and both precede the adoption of spelling norms. Middle French is more regular than Old French in some respects such as word order (Marchello-Nizia et al., 2020) and less in others such as NP structure and pronouns system (Marchello-Nizia, 1979). Medieval French covers a set of *Oïl* Romance languages spoken in the kingdom of France between the 9th and the 15th century (fig. 1.1).[8] There are around twenty such languages.

Older texts are close to Late Latin, and verse is prevalent until the end of the 13th century. Old French has a relatively free word order. Until the mid-11th century, the prevalent order is *Subject-Object-Verb* (SOV), which is then gradually supplanted by SVO, which is the default order in Contemporary French. Unlike most languages with free word order, the functions of verbal arguments are not always given away by morphological clues, the already simplistic case system of Old French disappears progressively through the covered period.



Figure 1.1: Oïl languages

[8] Hand-drawn figure by Mathilde Regnault.

There are also many cases of syntactic ambiguity. For example, in the following quote from *Lancelot*,[9] (verse 5436), both *la dame* and *Lancelot* could be the subject or the object of *Vit* and only the context enables the reader to understand that *la dame* is the subject.

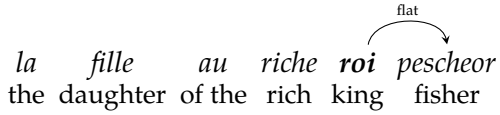
Dolant et pansif Lancelot Vit la dame
Mournful and meditative Lancelot saw the lady
'The lady saw that Lancelot was mournful and meditative.'

Word order is also relatively free within constituents. For example, a noun modifier can be on the left or on the right of its governor, and it is not necessarily preceded by a preposition. In contemporary French, it can only appear on the right, and it is found without a preposition only in some cases like named entities. Because of the general free word order and the absence of punctuation in our treebank, this adds up to the ambiguity of the analysis.

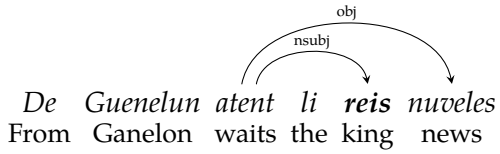
In each of the following examples from the SRCMF corpus (Prévost and Stein, 2013), the noun following *roi* ("king") has a different analysis: head of *roi*, modifier, argument of the same verb or a different one, with no explicit marking:



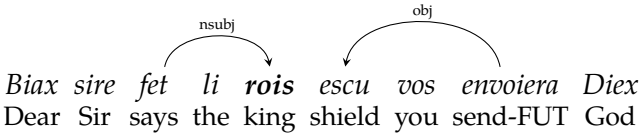
"Then were you not at the king's court anymore?" (*Beroul Tristan*)



"the daughter of the rich Fisher King" (*Queste del Saint Graal*)



"The king waits for news from Ganelon." (*Chanson de Roland*)



"Dear Sir, says the king, God will send you a shield." (*Queste del Saint Graal*)

[9] In the edition from Pierre Kunstmann, from the online *Base de français médiéval*: http://catalog.bfm-corpus.org/CharretteKu.

Furthermore, overt subjects are not mandatory, and are often dropped in texts written in verse until the 12th century, after which the presence of subjects increases through time. These phenomena are particularly prevalent in verse, where metric and rhyming constraints often lead to more contrived syntactic forms than in prose.

Another source of ambiguity is the variety of spellings, due to the lack of spelling standard. For example, the word *moult* (transl. *a lot (of), very*), emblematic of this period, is initially an adjective, and it is progressively grammaticalized, becoming an adverb. Several forms appear at the same time, some with a declension, some without, and the radical does not have a fixed spelling: *molt(e)(s)*, *molz*, *mult(e)(s)*, *mul(t)z*, *mou(l)t*...

1.3.2 Early Modern French

Source	Normalized	Translation
Surquoy, SIRE, s'il plaiſt à voſtre Maieſté de ſe ſouuenir des miſeres de ſon Eſtat, dõt au moins ell'a tiré cét aduantage, qu'en vne grande ieuneſse ell'a acquis vne grande experiẽce, elle verra que tous les malheurs de sõ bas âge ont pris leur commencement en ſemblables occaſions;	*Sur quoi, SIRE, s'il plaît à votre Majesté de se souvenir des misères de son état dont au moins elle a tiré cet avantage, qu'en une grande jeunesse elle a acquis une grande expérience, elle verra que tous les malheurs de son bas âge ont pris leur commencement en semblables occasions ;*	"Whereupon, SIR, if it pleases your Majesty to remember the miseries of her state, from which at least she has derived this advantage, that in great youth she has acquired great experience, she will see that all the misfortunes of her early life took their beginning on similar occasions;"

Table 1.1: Example of normalization taken from the *Lettres* of Guez de Balzac (1624).

We loosely define Early Modern French as a state of language following Middle French in 1500—following here the *terminus ad quem* used by the *Dictionnaire de Moyen Français* (Martin, Robert (dir.), 2020)—and ending with the French Revolution in 1789. It therefore encompasses three centuries (16th, 17th and 18th c.), or two linguistic periods: the *français préclassique* or "preclassical French", 1500–1630 and the *français classique* or "classical French", 1630–1689; both periodizations are currently used in French linguistics (*e.g.* by Vachon (2010) and Amatuzzi et al. (2019)).

A typical example of Early Modern French, taken from Guez de Balzac (1624), is given in Table 1.1. We note here the presence of several phenomena that have now disappeared in contemporary French, such as the presence of abbreviations (*dõt*→*dont*), the long *s* (*ſ*, see *miſeres*), the use of *v* instead of *u* (*vne* for *une*), the conservation of etymological letters (*voſtre* < Latin *vŏster* rather than *votre*) and calligraphic letters (*-y* in *Surquoy*), the absence of welding (*mal-heurs* and not *malheurs*) and the opposite (*Surquoy* and not *Sur quoi*).

For NLP systems, which process raw sequences, such differences with respect to contemporary French are not trivial, and they prevent the processing of historical texts with tools trained on recent sources.

1.4 Outline

This document is organized in 6 different parts, each of which comprises multiple chapters:

Part I Contains this introduction as well as two other chapters, one discussing the availability of raw textual resources throughout this thesis for both Contemporary and Historical languages; and the other discussing the state-of-the-art architectures that appeared during the course of this Ph.D., as well as a brief discussion of downstream tasks for contemporary French, the available datasets and the state-of-the-art models available for these tasks.

Part II Contains a detailed discussion, spanning from chapter 4 to chapter 8, of the curation, audits and evaluations, and improvements of our raw multilingual corpus OSCAR (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020b; Abadji et al., 2021; Abadji et al., 2022) intended for the pre-training of language models.

Part III Spans from chapter 9 to chapter 10, and concerns the curation of monolingual corpora, for both Contemporary and Historical French, intended for the pre-training of language models.

Part IV Comprises chapters 11 to 15, and concerns the pre-training and development of language models for Contemporary, Medieval and Early Modern French; as well as their respective evaluations across a wide range of downstream tasks.

Part V Contains chapter 16 which outlines the conclusions and perspectives of this Ph.D. thesis.

Part VI Finally, contains all the appendices as well as additional information that is relevant to our work.

2 On Raw Corpora for Language Modeling

In which the corpora available for the pre-training of language models both at the beginning and throughout this Ph.D. thesis are described. We also describe one of the pipelines used to produce one of the corpora, and at the end we discuss both the raw and annotated corpora available for some historical languages.

As previously stated, the only freely-available corpora, considered *large enough* for the pre-training of language models in Contemporary French at the beginning of this Ph.D thesis, were Wikipedia and frWac. The frWaC corpus (Baroni et al., 2009) is a French text corpus collected from the `.fr` domain with using medium-frequency words from the *Le Monde Diplomatique* corpus and basic French vocabulary lists as seeds. The corpus consists of French websites with total size 1.3 billion words. While frWac was almost 2 times bigger than the French Wikipedia at the time, it was still nowhere near the amount of data that was thought to be needed to properly train a Transformer-based language model at the time (Liu et al., 2019). However, we liked the idea of Baroni et al. (2009) of using web pages and crawling as a mean to obtain large quantities of textual data. This is why the work of Mikolov et al. (2018) and its use of Common Crawl and the FastText linear classifier (Joulin et al., 2016; Joulin et al., 2017), in order to gather large amounts of multilingual text plays a central role in the development of our own multilingual web based corpora.

In this chapter we first describe the work of Mikolov et al. (2018), then we present and discuss the large web corpora that became available through this thesis that were not available at the begging of it, and the challenges that came with them and with the ever-growing demand for large textual corpora. Finally, we briefly give an overview of the available corpora for historical languages.

2.1 Common Crawl

Common Crawl is a non-profit foundation which produces and maintains an open repository of web crawled data that is both accessible and analyzable.[1] Common

[1] http://commoncrawl.org/about/

Crawl's complete web archive consists of petabytes of data collected over 8 years of web crawling. The repository contains raw web page HTML data (WARC files), metadata extracts (WAT files) and plain text extracts (WET files). The organization's crawlers has always respected `nofollow`[2] and `robots.txt`[3] policies.

Each monthly Common Crawl snapshot is in itself a massive multilingual corpus, where every single file contains data coming from multiple web pages written in a large variety of languages and covering all possible types of topics. Thus, in order to effectively use this corpus for Natural Language Processing and Machine Learning applications, one has first to extract, filter, clean and classify the data in the snapshot by language.

Throughout this thesis, we will use the WET files which contain the extracted plain texts from the websites mostly converted to UTF-8, as well as headers containing the metadata of each crawled document. Each WET file comes compressed in gzip format[4] and is stored on Amazon Web Services.

Common Crawl has already been successfully used to train language models, even multilingual ones. The most notable example is probably FastText which was first trained for English using Common Crawl (Mikolov et al., 2018) and then for other 157 different languages (Grave et al., 2018). In fact Grave et al. (2018) proposed a pipeline to filter, clean and classify Common Crawl, which we shall call the "FastText pre-processing pipeline." They used the FastText linear classifier (Joulin et al., 2016; Joulin et al., 2017) to classify each line of Common Crawl by language, and downloaded the initial corpus and schedule the I/O using some simple Bash scripts. Their solution, however, proved to be a synchronous blocking pipeline that works well on infrastructures having the necessary hardware to assure high I/O speeds even when storing tens of terabytes of data at a time. But that downscales poorly to medium-low resource infrastructures that rely on more traditional cost-effective electromechanical mediums in order to store this amount of data.

2.1.1 FastText's Pipeline

The *"FastText pre-processing pipeline"* used by Grave et al. (2018) launches multiple process, preferably as many as available cores. Each of these processes first downloads one Common Crawl WET file which then proceeds to decompress after the download is over. After decompressing, an instance of the FastText linear classifier (Joulin et al., 2016; Joulin et al., 2017) is launched, the classifier processes each WET file line by line, generating a language tag for each line. The tags are then stored in a tag file which holds a one-to-one correspondence between lines of the WET file and its corresponding language tag. The WET file and the tag files are read sequentially and each on the WET file line holding the condition of being longer than 100 bytes is

[2] `http://microformats.org/wiki/rel-nofollow`
[3] `https://www.robotstxt.org/`
[4] `https://www.gnu.org/software/gzip/`

appended to a language file containing only plain text (tags are discarded). Finally, the tag file and the WET files are deleted.

Only when one of these processes finishes another can be launched. This means that one can at most process and download as many files as cores the machine has. That is, if for example a machine has 24 cores, only 24 WET files can be downloaded and processed simultaneously, moreover, the 25$^{\text{th}}$ file won't be downloaded until one of the previous 24 files is completely processed.

When all the WET files are classified, one would normally get around 160 language files, each file holding just plain text written in its corresponding language. These files still need to be filtered in order to get rid of all files containing invalid UTF-8 characters, so again a number of processes are launched, this time depending on the amount of memory of the machine. Each process reads a language file, first filters for invalid UTF-8 characters and then performs deduplication. A simple non-collision resistant hashing algorithm is used to deduplicate the files.

The FastText linear classifier works by representing sentences for classification as Bags of Words (BoW) and training a linear classifier. A weight matrix A is used as a look-up table over the words and the word representations are then averaged into a text representation which is fed to the linear classifier. The architecture is in general similar to the CBoW model of Mikolov et al. (2013), but the middle word is replaced by a label. They uses a softmax function f to compute the probability distribution over the classes. For a set of N documents, the model is trained to minimize the negative log-likelihood over the classes:

$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(f(BAx_n)),$$

where x_n is the normalized bag of features of the n-th document, y_n is the n-th label, and A, B are the weight matrices. The pre-trained FastText model for language recognition (Grave et al., 2018) is capable of recognizing around 176 different languages and was trained using 400 million tokens from Wikipedia as well as sentences from the Tatoeba website.[5]

2.1.2 CCNet

We note that the original Common-Crawl-based corpus created by Grave et al. (2018) to train FastText is not freely available. Shortly arfter having started working with Common Crawl data and developing a pipeline to classify it by language (Ortiz Suárez et al., 2019), a new architecture for creating a Common-Crawl-based corpus named CCNet (Wenzek et al., 2020) was published, although it included specialized filtering based on the KenLM library (Heafield, 2011) and trained on Wikipedia,

[5] https://tatoeba.org/

which might result in a cleaner corpus; the resulting CCNet corpus itself was never published in its entirety.

2.2 Demand for Large Corpora

Using large corpora to train neural language models dates back to way before the beginning of this Ph.D. thesis (Schwenk and Gauvain, 2005). During the course of our studies, we observed the demand for large corpora considerably increasing, specially the last two years with the advent of semi-supervised learning methods in NLP, in particular with *contextualized word representations* (Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019) and more recently *very large generative language models* like GPT-3, T5, GPT-Neo (Raffel et al., 2020; Brown et al., 2020; Black et al., 2021). While there have been some recent efforts to manually curate such corpora[6] (Gao et al., 2020), the common approach to collect large amounts of raw textual data still relies primarily on crawled web text (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020b; Xue et al., 2021; El-Kishky et al., 2020; Esplà et al., 2019; Bañón et al., 2020; Gao et al., 2020), and although some of the initial concerns of using crawled data (Trinh and Le, 2018; Radford et al., 2019) were addressed during the course of this particular Ph.D. thesis (Ortiz Suárez et al., 2020b; Martin et al., 2020) there a many concerns that still need to be tackled (Caswell et al., 2020) specially for multilingual data (Kreutzer et al., 2022).

In this demand for large raw textual corpora we can observe a clear back and forth in the type of data used to pre-train these models. On one hand some authors have opted for highly curated or edited data like Wikipedia such as Al-Rfou' et al. (2013) and Bojanowski et al. (2017) for static word embeddings, the 1B Word Benchmark (Chelba et al., 2014) for ELMo (Peters et al., 2018), and the BookCorpus (Zhu et al., 2015) and Wikipedia for BERT (Devlin et al., 2019). On the other hand projects like those of Pennington et al. (2014) or Grave et al. (2018) used crawled data for the pre-training of fixed word embeddings, CamemBERT (Martin et al., 2020), our contextualized model for French, successfully used only Crawled data for pre-training, and even large generative language models like T5 have used mainly crawled data successfully (Raffel et al., 2020). We can of course also see examples of projects successfully using a mix of both manually curated and automatically crawled data such as RoBERTa (Liu et al., 2019), FauBERT(Le et al., 2020b), XLNet (Yang et al., 2019) and GPT-Neo (Black et al., 2021; Gao et al., 2020). However, no matter the chosen approach to build these large corpora, there are in every case concerns that have been expressed, specially for the datasets used in very large generative language models (Bender et al., 2021), even when using manually edited resources like Wikipedia (Barera, 2020).

[6] https://bigscience.huggingface.co

2.2.1 Problems with Crawled Corpora

Corpora collected by web crawlers are known to be noisy (Junczys-Dowmunt, 2019; Luccioni and Viviano, 2021). In highly multilingual settings, past work found that web-crawls of lower-resource languages have serious issues, especially with segment-level LangID (Caswell et al., 2020). Cleaning and filtering web-crawls can boost general language modeling (Gao et al., 2020; Brown et al., 2020; Raffel et al., 2020) and downstream task performance (Moore and Lewis, 2010; Rarrick et al., 2011; Xu and Koehn, 2017; Khayrallah and Koehn, 2018; Brown et al., 2020).

As the scale of ML research grows, it becomes increasingly difficult to validate automatically collected and curated datasets (Biderman and Scheirer, 2020; Birhane and Prabhu, 2021; Bender et al., 2021). Several works have focused on advancing methodologies and best practices to address these challenges. Bender and Friedman (2018) introduced data statements, a documentary framework for NLP datasets that seeks to provide a universal minimum bar for dataset description. Similar work has been done on systematizing documentation in other areas in data science and machine learning, including work focusing on online news (Kevin et al., 2018), data ethics (Sun et al., 2019), and data exploration (Holland et al., 2018), as well as generalist work such as (Gebru et al., 2018). Data quality is also implicitly documented by successes of filtering methods. There is a large literature on filtering data for various NLP tasks, e.g. (Axelrod et al., 2011; Moore and Lewis, 2010; Rarrick et al., 2011; Wang et al., 2018; Kamholz et al., 2014; Junczys-Dowmunt, 2018; Caswell et al., 2020).

2.2.2 Publically Available Web-based Corpora

	Parallel			**Monolingual**	
	CCAligned	**ParaCrawl v7.1**	**WikiMatrix**	**OSCAR**	**mC4**
#languages	137	41	85	166	101
Source	CC 2013–2020	selected websites	Wikipedia	CC 11/2018	CC all
Filtering level	document	sentence	sentence	document	document
Langid	FastText	CLD2	FastText	FastText	CLD3
Alignment	LASER	Vec/Hun/BLEU-Align	LASER	-	-
Evaluation	TED-6	WMT-5	TED-45	POS/DEP-5	XTREME

Table 2.1: Comparison of parallel and monolingual corpora extracted from web documents, including their downstream evaluation tasks. All parallel corpora are evaluated for machine translation (BLEU). TED-6: `da`, `cr`, `sl`, `sk`, `lt`, `et`; TED-45: 45-language subset of (Qi et al., 2018); WMT-5: `cs`, `de`, `fi`, `lv`, `ro`. POS/DEP-5: part-of-speech labeling and dependency parsing for `bg`, `ca`, `da`, `fi`, `id`.

Table 2.1 provides an overview of the corpora of interest in this work. We selected the corpora for their multilinguality and the inclusion of understudied languages in

NLP. With the exception of WikiMatrix and ParaCrawl, all corpora are derived from Common Crawl (CC).[7]

CCAligned (El-Kishky et al., 2020) is a parallel dataset built off 68 CC snapshots. Documents are aligned if they are in the same language according to FastText LangID (Joulin et al., 2016; Joulin et al., 2017), and have the same URL but for a differing language code. These alignments are refined with cross-lingual LASER embeddings (Artetxe and Schwenk, 2019). For sentence-level data, they split on newlines and align with LASER, but perform no further filtering. Human annotators evaluated the quality of document alignments for six languages (`de`, `zh`, `ar`, `ro`, `et`, `my`)[8] selected for their different scripts and amount of retrieved documents, reporting precision of over 90%. The quality of the extracted parallel sentences was evaluated in a machine translation (MT) task on six European (`da`, `cr`, `sl`, `sk`, `lt`, `et`) languages of the TED corpus (Qi et al., 2018), where it compared favorably to systems built on crawled sentences from WikiMatrix and ParaCrawl v6.

Multilingual C4 (mC4) (Xue et al., 2021) is a document-level dataset used for training the mT5 language model. It consists of monolingual text in 101 languages and is generated from 71 CC snapshots. It filters out pages that contain less than three lines of at least 200 characters and pages that contain bad words.[9] Since this is a document-level dataset, we split it by sentence and deduplicate it before rating. For language identification, it uses CLD3 (Botha et al., 2017),[10] a small feed-forward neural network that was trained to detect 107 languages. The mT5 model pre-trained on mC4 is evaluated on 6 tasks of the XTREME benchmark (Hu et al., 2020) covering a variety of languages and outperforms other multilingual pre-trained language models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020).

OSCAR (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020b) our own corpus, to which we devote the whole Part II of this Ph.D thesis, is a set of monolingual corpora extracted from CC snapshots, specifically from the plain text *WET* format distributed by CC which removes all the HTML tags and converts the text to UTF-8. It is deduplicated and follows the approach by (Grave et al., 2018) of using FastText LangID (Joulin et al., 2016; Joulin et al., 2017) on a line-level.[11,12] No other filtering was applied. For five languages (`bg`, `ca`, `da`, `fi`, `id`) OSCAR was used by its original authors to train language models which were then evaluated on parsing and POS

[7] http://commoncrawl.org/

[8] For language codes please refer to the IETF BCP 47 language tag https://www.iana.org/assignments/language-subtag-registry/language-subtag-registry

[9] https://github.com/LDNOOBW/

[10] https://github.com/google/cld3/

[11] Line-level here refers to the '\n' separated lines in the Common Crawl WET files.

[12] https://fasttext.cc/docs/en/language-identification.html

tagging (Ortiz Suárez et al., 2020b). OSCAR has also been used in independent studies to train monolingual or multilingual language models (`ar`, `as`, `bn`, `de`, `el`, `fr`, `gu`, `he`, `hi`, `kn`, `ml`, `mr`, `nl`, `or`, `pa`, `ro`, `ta`, `te`) and subsequently evaluate them on various downstream tasks (Antoun et al., 2021; Kakwani et al., 2020; Wilie et al., 2020; Chan et al., 2020; Koutsikakis et al., 2020; Martin et al., 2020; Chriqui and Yahav, 2021; Seker et al., 2021; Delobelle et al., 2020; Dumitrescu et al., 2020; Masala et al., 2020).

ParaCrawl v7.1 is a parallel dataset with 41 language pairs primarily aligned with English (39 out of 41) and mined using the parallel-data-crawling tool Bitextor (Esplà et al., 2019; Bañón et al., 2020) which includes downloading documents, preprocessing and normalization, aligning documents and segments, and filtering noisy data via Bicleaner.[13] ParaCrawl focuses on European languages, but also includes 9 lower-resource, non-European language pairs in v7.1. Sentence alignment and sentence pair filtering choices were optimized for five languages (`mt`, `et`, `hu`, `cs`, `de`) by training and evaluating MT models on the resulting parallel sentences. An earlier version (v5) was shown to improve translation quality on WMT benchmarks for `cs`, `de`, `fi`, `lv`, `ro`.

WikiMatrix (Schwenk et al., 2021) is a public dataset containing 135M parallel sentences in 1620 language pairs (85 languages) mined from Wikipedia. Out of the 135M parallel sentences, 34M are aligned with English. The text is extracted from Wikipedia pages, split into sentences, and duplicate sentences are removed. FastText LangID is used before identifying bitext with LASER's distance-based mining approach. The margin threshold is optimized by training and evaluating downstream MT models on four WMT benchmarks (`de-en`, `de-fr`, `cs-de`, `cs-fr`). The final dataset is used to train translation models that are then evaluated by automatically measuring the quality of their translations against human translations of TED talks in 45 languages, with highest quality for translations between English and e.g. `pt`, `es`, `da`, and lowest for `sr`, `ja`, `mr`, `zh_TW`.

2.3 Historical French Corpora

Large datasets for historical states of languages or extinct languages do exist. The *Corpus Middelnederlands* for Medieval Dutch (Reenen, Pieter van and Mulder, Maaike, 1998) and the *Base Geste* for Medieval French (Jean-Baptiste-Camps et al., 2019) are freely available online, encoded in TEI. It is also the case for other corpora for later states of language, such as the *Reference corpus of historical Slovene*, covering approximately three centuries of Slovene (1584–1899) (Erjavec, 2015), and the "corpus noyau" of *Presto* (Blumenthal and Vigier, 2018). This last corpus, in its extended version, uses other French corpora such as *Espistemon* for Renaissance French (Demonet,

[13] https://github.com/bitextor/bicleaner

1998–) and the University of Chicago's *American and French Research on the Treasury of the French Language* (ARTFL) (Morrissey and Olsen, 1981–); or like Frantext (ATILF, 1998–b), which is a generalist French corpus, covering the different states of the French language between the 11th and the 21st century. Although most of these text collections are free, the two biggest ones, Frantext and ARTFL, are not freely available or open-sourced.

Regarding corpora annotated corpora for historical languages, very few of them have manually annotated syntactical resources for their medieval states. English has three such treebanks (Archive, 2001; Kroch et al., 2000; Traugott and Pintzuk, 2008) for Old and Middle English. The TOROT treebank for Old Church Slavonic, Old East Slavonic and Middle Russian is another large resource (Berdicevskis and Eckhoff, 2020). There is a treebank for Medieval Latin as well, the *Index Thomisticus Treebank* (Passarotti, 2019). To our knowledge, the last large treebank containing medieval texts is IcePaHC for Icelandic (Rögnvaldsson et al., 2012). Some other corpora were annotated automatically in order to reduce the cost of annotation. For example, Rocio et al. (2003) adapted a parsing pipeline for contemporary Portuguese and Lee and Kong (2014) used a previously annotated treebank (Lee and Kong, 2012) to parse a larger medieval Chinese corpus. Concerning contemporary regional Romance languages, Miletic et al. (2020) also used a smaller treebank to generate new annotations, and concluded that using similar languages to train a model does not improve parsing. Although there are many resources for Latin, and some for Ancient Greek, we do not include them here, because they do not face the same challenges as medieval states of language, in particular the high level of spelling variability. And of course for Medieval French there is the SRCMF treebank that will be extensively used in chapter 14.

3 On Language Models and Downstream Tasks

In which we give a brief overview of word representations, we present a few of the non-English representations that were available when we began this Ph.D. thesis. We also give an overview of the downstream tasks that we will use for evaluation as well as brief description of the state-of-the-art models that were available when we started working on these tasks. We finally give an overview of the available neural models for historical languages.

Language models and more specifically word representations take a central part in the approach that we have chosen to tackle the tasks laid by the BASNUM project. In this chapter we give an overview of the available word representations that were available we began this Ph.D. thesis as well as a small description of the architecture of said representations.

We will also discuss the downstream tasks that we will use for evaluation focusing on particular on what was available for the French language throughout the course of this Ph.D. thesis. We also give briefly discuss the state-of-the-art models and studies for these tasks. Finally, we end the chapter by discussing some advancements in neural models for historical languages.

3.1 On Word Representations

One of the key elements that has pushed the state of the art considerably in neural NLP in recent years has been the introduction and spread of transfer learning methods to the field. These methods can normally be classified in two categories according to how they are used:

- *Feature-based* methods, which involve pre-training real-valued vectors ("embeddings") at the word, sentence, or paragraph level; and using them in conjunction with a specific architecture for each individual downstream task.

- *Fine-tuning* methods, which introduce a minimal number of task-specific parameters, and instead copy the weights from a pre-trained network and then tune them to a particular downstream task.

Embeddings or language models can be divided into *fixed*, meaning that they generate a single representation for each word in the vocabulary; and *contextualized*, meaning that a representation is generated based on both the word and its surrounding context, so that a single word can have multiple representations, each one depending on how it is used.

In practice, most fixed embeddings are used as feature-based models. The most notable examples are *word2vec* (Mikolov et al., 2013), *GloVe* (Pennington et al., 2014) and *FastText* (Mikolov et al., 2018). All of them are extensively used in a variety of applications nowadays. On the other hand, contextualized word representations and language models have been developed using both feature-based architectures, the most notable examples being ELMo and Flair (Peters et al., 2018; Akbik et al., 2018), and transformer based architectures, that are commonly used in a fine-tune setting, as is the case of GPT-1, GPT-2 (Radford et al., 2018, 2019), BERT and its derivatives (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020) and more recently XLNet (Yang et al., 2019) and T5 (Raffel et al., 2020). All of them have repeatedly improved the state-of-the art in many downstream NLP tasks over the last year.

In general, the main advantage of using language models is that they are mostly built in an *unsupervised* manner, and they can be trained with raw, unannotated plain text. However, their main drawback is that enormous quantities of data seem to be required to properly train them especially in the case of contextualized models, for which larger corpora are thought to be needed to properly address polysemy and cover the wide range of uses that commonly exist within languages.

For the first English version of word2vec, Mikolov et al. (2013) used a one billion word dataset consisting of various news articles. Later Al-Rfou' et al. (2013) and then Bojanowski et al. (2017) used the plain text from Wikipedia to train distributions of word2vec and FastText respectively, for languages other than English. Now, the problem of obtaining large quantities of data aggravates even more for contextual models, as they normally need multiple instances of a given word in order to capture all its different uses and in order to avoid overfitting due to the large quantity of hyperparameters that these models have. Peters et al. (2018) for example use a 5.5 billion token[1] dataset comprised of crawled news articles plus the English Wikipedia in order to train ELMo, Devlin et al. (2019) use a 3.3 billion word[2] corpus made by merging the English Wikipedia with the BooksCorpus (Zhu et al., 2015), and Radford et al. (2019) use a 40GB English corpus created by scraping outbound links from Reddit.[3]

[1] Punctuation marks are counted as tokens.
[2] Space sparated tokens.
[3] https://www.reddit.com/

For gathering data in a wide range of languages, Wikipedia is a commonly used option. It has been used to train fixed embeddings (Al-Rfou' et al., 2013; Bojanowski et al., 2017) and more recently the multilingual BERT (mBERT) (Devlin et al., 2019). However, for some languages, Wikipedia might not be large enough to train good quality contextualized word embeddings. Moreover, Wikipedia data all belong to the same specific genre and style. To address this problem, one can resort to crawled text from the internet; the largest and most widespread dataset of crawled text being Common Crawl.[4] Such an approach generally solves the quantity and genre/style coverage problems but might introduce noise in the data, an issue which has earned the corpus some criticism, most notably by Trinh and Le (2018) and Radford et al. (2019). Using Common Crawl also leads to data management challenges as the corpus is distributed in the form of a large set of plain text each containing a large quantity of unclassified multilingual documents from different websites.

Concerning contextual models, Baevski et al. (2019) trained a BERT-like model for English using Common Crawl. They followed the "fastText pre-processing pipeline" but they removed all copies of Wikipedia inside Common Crawl. They also trained their model using News Crawl (Bojar et al., 2018) and using Wikipedia + BooksCorpus, they compared three models and showed that Common Crawl gives the best performance out of the three corpora in spite of the previously mentioned criticism.

The XLNet model was trained for English by joining the BookCorpus, English Wikipedia, Giga5 (Parker et al., 2011), ClueWeb 2012-B (Callan et al., 2009) and Common Crawl. Particularly for Common Crawl, Yang et al. (2019) say they use "heuristics to aggressively filter out short or low-quality articles" from Common Crawl, however they don't give any detail about these "heuristics" nor about the pipeline they use to classify and extract the English part of Common Crawl. It is important to note that none of these projects distributed their classified, filtered and cleaned versions of Common Crawl, making it difficult in general to faithfully reproduce their results.

3.1.1 Some Language Model Architectures

ELMo: Contextualized word embeddings *Embeddings from Language Models* (ELMo) (Peters et al., 2018) is a Language Model, i.e, a model that given a sequence of N tokens, $(t_1, t_2, ..., t_N)$, computes the probability of the sequence by modeling the probability of token t_k given the history $(t_1, ..., t_{k-1})$:

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_1, t_2, \dots, t_{k-1}).$$

[4] https://commoncrawl.org

More precisely, ELMo uses a bidirectional language model, which combines a forward and a backward LSTM-based language model. ELMo also computes a context-independent token representation via a CNN over characters.

BERT and RoBERTa BERT (Devlin et al., 2019) is a multi-layer bidirectional Transformer encoder trained with a masked language modeling (MLM) objective, inspired by the Cloze task (Taylor, 1953). It comes in two sizes: the $\text{BERT}_{\text{BASE}}$ architecture and the $\text{BERT}_{\text{LARGE}}$ architecture. The $\text{BERT}_{\text{BASE}}$ architecture is 3 times smaller and therefore faster and easier to use while $\text{BERT}_{\text{LARGE}}$ achieves increased performance on downstream tasks. RoBERTa (Liu et al., 2019) improves the original implementation of BERT by identifying key design choices for better performance, using dynamic masking, removing the next sentence prediction task, training with larger batches, on more data, and for longer.

3.2 Non-English Word Representations

Since the introduction of *word2vec* (Mikolov et al., 2013), many attempts have been made to create multilingual language representations; for fixed word embeddings the most remarkable works are those of (Al-Rfou' et al., 2013) and (Bojanowski et al., 2017) who created word embeddings for a large quantity of languages using Wikipedia, and later (Grave et al., 2018) who trained the FastText word embeddings for 157 languages using Common Crawl and who in fact showed that using crawled data significantly increased the performance of the embeddings especially for mid- to low-resource languages.

Regarding contextualized models, the most notable non-English contribution has been that of the mBERT (Devlin et al., 2019), which is distributed as (i) a single multilingual model for 100 different languages trained on Wikipedia data, and as (ii) a single multilingual model for both Simplified and Traditional Chinese. Four monolingual fully trained ELMo models have been distributed for Japanese, Portuguese, German and Basque;[5] 44 monolingual ELMo models[6] where also released by the *HIT-SCIR* team (Che et al., 2018) during the *CoNLL 2018 Shared Task* (Zeman et al., 2018), but their training sets where capped at 20 million words.

Following the success of large pre-trained language models, they were extended to the multilingual setting with multilingual BERT (hereafter mBERT) (Devlin et al., 2019), a single multilingual model for 104 different languages trained on Wikipedia data, and later XLM (Conneau and Lample, 2019), which significantly improved unsupervized machine translation. More recently XLM-R (Conneau et al., 2020), extended XLM by training on 2.5TB of data and outperformed previous scores on multilingual benchmarks. They show that multilingual models can obtain results

[5] https://allennlp.org/elmo
[6] https://github.com/HIT-SCIR/ELMoForManyLangs

competitive with monolingual models by leveraging higher quality data from other languages on specific downstream tasks.

A few non-English monolingual models had been released at the beginning of this thesis: ELMo models for Japanese, Portuguese, German and Basque[7] and BERT for Simplified and Traditional Chinese (Devlin et al., 2019) and German (Chan et al., 2019).

However, to the best of our knowledge, at the beginning of this Ph.D. thesis, no particular effort had been made toward training models for languages other than English at a scale similar to the latest English models (e.g. RoBERTa trained on more than 100GB of data).

3.2.1 Recent Developments in Contextualized Representations

Since the introduction of contextualized word representations (Peters et al., 2018; Akbik et al., 2018; Devlin et al., 2019) and the many improvements proposed for them in the consumption of computational resources (Clark et al., 2020), in the amount of data required to fine-tune them (Raffel et al., 2020), and more recently in the length of the contextual window (Xiong et al., 2021); there have also been important advancements from a digital humanities point of view on *unsupervised domain adaptation* (Ramponi and Plank, 2020). In this case, one specializes a language model to a particular domain with unlabeled data in order to improve performance in downstream tasks. This can be achieved by pre-training the models from scratch with specialized data (Beltagy et al., 2019) or by continuing the training of a general model with a new corpus (Lee et al., 2019; Peng et al., 2019). This last method has already been successfully implemented in the context of historical languages, in particular Han and Eisenstein (2019) showed that one can successfully adapt the original BERT (Devlin et al., 2019) to Early Modern English by continuing the pre-training on historical raw texts.

In a multilingual context, transformer-based models such as mBERT have been adapted to low-resource languages and evaluated in dependency parsing and POS-tagging showing promising results (Chau et al., 2020; Muller et al., 2021; Gururangan et al., 2020; Wang et al., 2020b). However, this multilingual approach has also been criticized for favoring monolingual pre-training even when data is scarce (Virtanen et al., 2019; Ortiz Suárez et al., 2020b). Indeed, even when only small pre-training corpora are available, BERT-like models have also been successfully pre-trained, resulting in well-performing models (Micheli et al., 2020). Furthermore, compact BERT-like models have also been studied (Turc et al., 2019) and might prove useful in data constrained conditions, such as monolingual pre-training of contextualized word representation for low-resource languages. In any case BERT-Like models have

[7] https://allennlp.org/elmo

been one of the most prolific subjects of study in NLP since the start of this Ph.D. thesis (Rogers et al., 2020).

3.3 Downstream Tasks Evaluation

POS-Tagging and Dependency Parsing POS-tagging is a low-level syntactic task, which consists in assigning to each word its corresponding grammatical category. Dependency-parsing consists of higher order syntactic task like predicting the labeled syntactic tree capturing the syntactic relations between words. We evaluate the performance of our models throughout this text, by using the standard UPOS accuracy for POS-tagging, and Unlabeled Attachment Score (UAS) and Labeled Attachment Score (LAS) for dependency parsing. We will always assume gold tokenization and gold word segmentation as provided in the UD treebanks.

For both of these tasks we always run our experiments using the Universal Dependencies (UD)[8] framework and its corresponding UD POS tag set (Petrov et al., 2012) and UD treebank collection (Nivre et al., 2018), which was used for the CoNLL 2018 shared task (Seker et al., 2018). We often perform the evaluations of our Contemporary French models on the four freely available French UD treebanks in UD v2.2: GSD (McDonald et al., 2013), Sequoia[9] (Candito and Seddah, 2012; Candito et al., 2014), Spoken (Lacheret et al., 2014; Bawden et al., 2014),[10] and ParTUT (Sanguinetti and Bosco, 2015).

Natural Language Inference (NLI) We will further evaluate some of our models on NLI, using the French part of the XNLI dataset (Conneau et al., 2018). NLI consists in predicting whether a hypothesis sentence is entailed, neutral or contradicts a premise sentence. The XNLI dataset is the extension of the Multi-Genre NLI (MultiNLI) corpus (Williams et al., 2018) to 15 languages by translating the validation and test sets manually into each of those languages. The English training set is machine translated for all languages other than English. The dataset is composed of 122k train, 2490 development and 5010 test examples for each language. As usual, NLI performance is evaluated using accuracy.

Named Entity Recognition (NER) Finally, we also evaluate our models in NER, which is a sequence labeling task predicting which words refer to real-world objects, such as people, locations, artifacts and organizations. We use the French Treebank[11] (FTB) (Abeillé et al., 2003) in its 2008 version introduced by Candito and Crabbé

[8] https://universaldependencies.org
[9] https://deep-sequoia.inria.fr
[10] Speech transcript uncased that includes annotated disfluencies without punctuation
[11] This dataset has only been stored and used on Inria's servers after signing the research-only agreement.

(2009) and with NER annotations by Sagot et al. (2012). The FTB contains more than 11 thousand entity mentions distributed among 7 different entity types. A brief overview of the FTB can also be found in Table 3.1. As NER plays a central role in this Ph.D. thesis, beign the task in which we evaluate our models the most, we give an expanded overview of the task in the following paragraphs.

Named Entity Recognition

Named entity recognition (NER) is the widely studied task consisting in identifying text spans that denote *named entities* such as person, location and organization names, to name the most important types. Such text spans are called named entity *mentions*. In NER, mentions are generally not only identified, but also classified according to a more or less fine-grained ontology, thereby allowing for instance to distinguish between the telecommunication company *Orange* and the town *Orange* in southern France (among others). Importantly, it has long been recognized that the type of named entities can be defined in two ways, which underlies the notion of metonymy: the intrinsic type (*France* is always a location) and the contextual type (in *la France a signé un traité* 'France signed a treaty', *France* denotes an organization).

NER has been an important task in natural language processing for quite some time. It was already the focus of the MUC conferences and associated shared tasks (Marsh and Perzanowski, 1998), and later that of the CoNLL 2003 and ACE shared tasks (Tjong Kim Sang and De Meulder, 2003; Doddington et al., 2004). Traditionally, as for instance was the case for the MUC shared tasks, only person names, location names, organization names, and sometimes "other proper names" are considered. However, the notion of named entity mention is sometimes extended to cover any text span that does not follow the general grammar of the language at hand, but a type- and often culture-specific grammar, thereby including entities ranging from product and brand names to dates and from URLs to monetary amounts and other types of numbers.

As for many other tasks, NER was first addressed using rule-based approaches, followed by statistical and now neural machine learning techniques (see Section 3.3.3 for a brief discussion on NER approaches). Of course, evaluating NER systems as well as training machine-learning-based NER systems, statistical or neural, require named-entity-annotated corpora. Unfortunately, most named entity annotated French corpora are oral transcripts, and they are not always freely available. The ESTER and ESTER2 corpora (60 plus 150 hours of NER-annotated broadcast transcripts) (Galliano et al., 2005, 2009), as well as the Quaero (Grouin et al., 2011) corpus are based on oral transcripts (radio broadcasts). Interestingly, the Quaero corpus relies on an original, very rich and structured definition of the notion of named entity (Rosset et al., 2011). It contains both the intrinsic and the contextual types of each mention, whereas the ESTER and ESTER2 corpora only provide the contextual type.

3.3.1 Evaluation Datasets

Treebanks As previously stated, we perform most our evaluations on the four freely available French UD treebanks in UD v2.2: GSD, Sequoia, Spoken, and ParTUT, presented in Table 3.1.

GSD treebank (McDonald et al., 2013) is the second-largest tree-bank available for French after the FTB (described in subsection 3.3.1), it contains data from blogs, news, reviews, and Wikipedia.

Sequoia tree-bank (Candito et al., 2014) comprises more than 3000 sentences, from the French Europarl, the regional newspaper *L'Est Républicain*, the French Wikipedia and documents from the European Medicines Agency.

Spoken was automatically converted from the Rhapsodie tree-bank (Lacheret et al., 2014) with manual corrections. It consists of 57 sound samples of spoken French with phonetic transcription aligned with sound (word boundaries, syllables, and phonemes), syntactic and prosodic annotations.

ParTUT is a conversion of a multilingual parallel treebank developed at the University of Turin, and consisting of a variety of text genres, including talks, legal texts, and Wikipedia articles, among others; ParTUT data is derived from the already-existing parallel treebank, Par(allel)TUT (Sanguinetti and Bosco, 2015). Table 3.1 contains a summary comparing the sizes of the treebanks.

Treebank	#Tokens	#Sentences	Genres
GSD	389,363	16,342	Blogs, News Reviews, Wiki
Sequoia	68,615	3,099	Medical, News Non-fiction, Wiki
Spoken	34,972	2,786	Spoken
ParTUT	27,658	1,020	Legal, News, Wikis
FTB	350,930	27,658	News

Table 3.1: Statistics on the treebanks used in POS tagging, dependency parsing, and NER (FTB).

Again as we pay special attention to the NER task, and we actually work directly on the FTB by aligning both the Universal Dependencies and the TEI-annotated NER version of it in section 9.2.1, we provide an expanded overview of the corpus here below.

The original named entity FTB annotation layer

Sagot et al. (2012) annotated the FTB with the span, absolute type,[12] sometimes subtype and Aleda unique identifier of each named entity mention.[13] Annotations are restricted to person, location, organization and company names, as well as a few product names.[14] There are no nested entities. Non capitalized entity mentions (e.g. *banque mondiale* 'World Bank') are annotated only if they can be disambiguated independently of their context. Entity mentions that require the context to be disambiguated (e.g. *Banque centrale*) are only annotated if they are capitalized.[15] For person names, grammatical or contextual words around the mention are not included in the mention (e.g. in *M. Jacques Chirac* or *le Président Jacques Chirac*, only *Jacques Chirac* is included in the mention).

Tags used for the annotation have the following information:

- the identifier of the NE in the free large-scale entity database for French Aleda database (Sagot and Stern, 2012) (`eid` attribute); when a named entity is not present in the database, the identifier is `null`,[16]
- the normalized named of the named entity as given in Aleda; for locations it is their name as given in GeoNames and for the others, it is the title of the corresponding French Wikipedia article,
- the type and, when relevant, the subtype of the entity.

Here are two annotation examples:

```
<ENAMEX type="Organization" eid="1000000000016778" name="Confédération
française démocratique du travail">CFDT</ENAMEX>
<ENAMEX type="Location" sub_type="Country" eid="2000000001861060"
name="Japan">Japon</ENAMEX>
```

Sagot et al. (2012) annotated the 2007 version of the FTB treebank (with the exception of sentences that did not receive any functional annotation), i.e. 12,351 sentences comprising 350,931 tokens. The annotation process consisted in a manual

[12] Every mention of *France* is annotated as a `Location` with subtype `Country`, as given in Aleda (a free large-scale entity database for French) database (Sagot and Stern, 2012), even if in context the mentioned entity is a political organization, the French people, a sports team, etc.

[13] Only proper nouns are considered as named entity mentions, thereby excluding other types of referential expressions.

[14] More precisely, we used a tagset of 7 base NE types: `Person`, `Location`, `Organization`, `Company`, `Product`, `POI` (Point of Interest) and `FictionChar`.

[15] So for instance, in *université de Nantes* 'Nantes university', only *Nantes* is annotated, as a city, as *université* is written in lowercase letters. However, *Université de Nantes* 'Nantes University' is wholly annotated as an organization. It is non-ambiguous because *Université* is capitalized. *Université de Montpellier* 'Montpellier University' being ambiguous when the text of the FTB was written and when the named entity annotations were produced, only *Montpellier* is annotated, as a city.

[16] Specific conventions for entities that have merged, changed name, ceased to exist as such (e.g. *Tchequoslovaquie*) or evolved in other ways are described in Sagot et al. (2012).

correction and validation of the output of a rule- and heuristics-based named entity recognition and linking tool in an XML editor. Only a single person annotated the corpus, despite the limitations of such a protocol, as acknowledged by Sagot et al. (2012).

In total, 5,890 of the 12,351 sentences contain at least a named entity mention. 11,636 mentions were annotated, which are distributed as follows: 3,761 location names, 3,357 company names, 2,381 organization names, 2,025 person names, 67 product names, 29 fiction character names and 15 points of interest.

3.3.2 Brief State of the Art for Dependency Parsing and POS Tagging

For dependency parsing and POS tagging the most notable non-English specific contribution is that of the *CoNLL 2018 Shared Task* (Zeman et al., 2018), where the 1st place (LAS Ranking) was awarded to the *HIT-SCIR* team (Che et al., 2018) who used Dozat and Manning (2017)'s *Deep Bi-affine parser* and its extension described in (Dozat et al., 2017), coupled with deep contextualized ELMo embeddings (Peters et al., 2018) (capping the training set at 20 million words). The 1st place in universal POS tagging was awarded to Smith et al. (2018) who used two separate instances of Bohnet et al. (2018)'s tagger.

More recent developments in POS tagging and parsing include those of Straka et al. (2019) which couples another CoNLL 2018 shared task participant, UDPipe 2.0 (Straka, 2018), with mBERT greatly improving the scores of the original model, and UDify (Kondratyuk and Straka, 2019), which adds an extra attention layer on top of mBERT plus a Deep Bi-affine attention layer for dependency parsing and a Softmax layer for POS tagging. UDify is actually trained by concatenating the training sets of 124 different UD treebanks, creating a single POS tagging and dependency parsing model that works across 75 different languages.

3.3.3 Brief State of the Art for NER

As mentioned above, NER was first addressed using rule-based approaches, followed by statistical and now neural machine learning techniques. In addition, many systems use a lexicon of named entity mentions, usually called a "gazetteer" in this context.

Most of the advances in NER have been achieved on English, in particular with the CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) and Ontonotes v5 (Pradhan et al., 2012, 2013) corpora. In recent years, NER was traditionally tackled using Conditional Random Fields (CRF) (Lafferty et al., 2001) which are quite suited for NER; CRFs were later used as decoding layers for Bi-LSTM architectures (Huang et al., 2015; Lample et al., 2016) showing considerable improvements over CRFs alone. These Bi-LSTM-CRF architectures were later enhanced with contextualized word embeddings which yet again brought major improvements to the task (Peters et al., 2018; Akbik et al., 2018). Finally, large pre-trained architectures settled the

current state of the art showing a small yet important improvement over previous NER-specific architectures (Devlin et al., 2019; Baevski et al., 2019).

For French, rule-based system have been developed until relatively recently, due to the lack of proper training data (Sekine and Nobata, 2004; Rosset et al., 2005; Stern and Sagot, 2010; Nouvel et al., 2014). The limited availability of a few annotated corpora made it possible to apply statistical machine learning techniques (Béchet and Charton, 2010; Dupont and Tellier, 2014; Dupont, 2017) as well as hybrid techniques combining handcrafted grammars and machine learning (Béchet et al., 2011). To the best of our knowledge, the best results previously published on FTB NER are those obtained by Dupont (2017), who trained both CRF and BiLSTM-CRF architectures and improved them using heuristics and pre-trained word embeddings.

Leaving aside French and English, the CoNLL 2002 shared task included NER corpora for Spanish and Dutch corpora (Tjong Kim Sang, 2002) while the CoNLL 2003 shared task included a German corpus (Tjong Kim Sang and De Meulder, 2003). The recent efforts by Straková et al. (2019) settled the state of the art for Spanish and Dutch, while Akbik et al. (2018) did so for German.

3.3.4 Common Baselines

In dependency parsing and POS-tagging we will often compare our models with:

- *mBERT*: The multilingual cased version of BERT. We often fine-tune mBERT on each of the treebanks with an additional layer for POS-tagging and dependency parsing.

- $XLM_{MLM\text{-}TLM}$: A multilingual pretrained language model from Conneau and Lample (2019), which showed better performance than mBERT on NLI. We use the version available in the Hugging's Face transformer library (Wolf et al., 2019).

- *UDify* (Kondratyuk and Straka, 2019): A multitask and multilingual model based on mBERT, UDify is trained simultaneously on 124 different UD treebanks, creating a single POS tagging and dependency parsing model that works across 75 different languages. We always report the scores from Kondratyuk and Straka (2019) paper.

- *UDPipe Future* (Straka, 2018): An LSTM-based model ranked 3rd in dependency parsing and 6th in POS tagging at the CoNLL 2018 shared task (Seker et al., 2018). We always report the scores from Kondratyuk and Straka (2019) paper.

- *UDPipe Future + mBERT + Flair* (Straka et al., 2019): The original UDPipe Future implementation using mBERT and Flair as feature-based contextualized word embeddings. We always report the scores from Straka et al. (2019) paper.

In French, as discussed avobe, no extensive work has been done on NER due to the limited availability of annotated corpora. Thus, we compare our models with the only recent available baselines set by Dupont (2017), who trained both CRF (Lafferty et al., 2001) and BiLSTM-CRF (Lample et al., 2016) architectures on the FTB and enhanced them using heuristics and pretrained word embeddings. Additionally, as for POS and dependency parsing, we often compare our models to a fine-tuned version of mBERT for the NER task.

For XNLI, we provide the scores of mBERT which has been reported for French by Wu and Dredze (2019). We report scores from $\text{XLM}_{\text{MLM-TLM}}$ (described above), the best model from Conneau and Lample (2019). We will also report the results of XLM-R (Conneau et al., 2020).

3.4 Neural Models for Historical Languages

Efficient language models have been trained for less-resourced or extinct Languages such as Latin (Bamman and Burns, 2020), following the approach of Martin et al. (2020) for training language models with fewer data than was previously thought. There have also been some recent projects that specifically target Early Modern French such as that of Pie Extended (Clérice, 2020) that uses the hierarchical encoding architecture originally proposed by Manjavacas et al. (2019) which itself is constructed by stacking multiple Bi-LSTM-CRFs. Clérice (2020) distributes pre-trained models for POS tagging and lemmatisation.

Lastly, concerning dependency parsing and POS-tagging of Old French in particular, the works of Guibon et al. (2014) and Stein (2014, 2016) are noteworthy. However, they use very different approaches to the one used throughout this thesis and evaluate on previous versions of SRCMF, with incompatible annotation choices and slightly different texts. For the UD version of SRCMF, the most notable work is that of UDPipe 2.0 (Straka, 2018), which was later enhanced by including contextualized word embeddings (Straka et al., 2019).

Part II

OSCAR

4 Goclassy: an Asynchronous Language Classification Pipeline for Common Crawl

> In which we present the work of Ortiz Suárez et al. (2019), introducing the first OSCAR corpus, now known as *OSCAR 2019*, as well as asynchronous pipeline `goclassy` that was used to produce OSCAR 2019 and that was specifically conceived to be used in low resource infrastructures.[1]

As previously mentioned, back in the fall of 2018 when this Ph.D. started, there was no freely available contemporary French corpus of the size that was thought to be needed at that time in order to train a state-of-the art language model. The only available resources were the French Wikipedia and frWAC (Baroni et al., 2009). At the time, the original fastText's language classification pipeline (Grave et al., 2018) was recently published, but while Grave et al. (2018) published word embeddings for a wide range of languages using the produced corpus, the corpus itself was never published. We thus decided to reproduce and improve Grave et al. (2018) in order to get enough raw textual French data to train a language model. Given that our pipeline ended up being capable of classifying text in a wide range of languages, we decided to publish a multilingual corpus instead of a monolingual French one. In this chapter we thus lay the details of the `goclassy` as well as the first version of the OSCAR corpus.

4.1 An Asynchronous Pipeline

We propose a new pipeline derived from the fastText one which we call `goclassy`, we reuse the fastText linear classifier (Joulin et al., 2016; Joulin et al., 2017) and the pre-trained fastText model for language recognition (Grave et al., 2018), but we completely rewrite and parallelize their pipeline in an asynchronous manner.

The order of operations is more or less the same as in the fastText pre-processing pipeline but instead of clustering multiple operations into a single blocking process,

[1] Contributions: I did the concrete work and designed most of it under the supervision of my coauthors and Ph.D. supervisors.

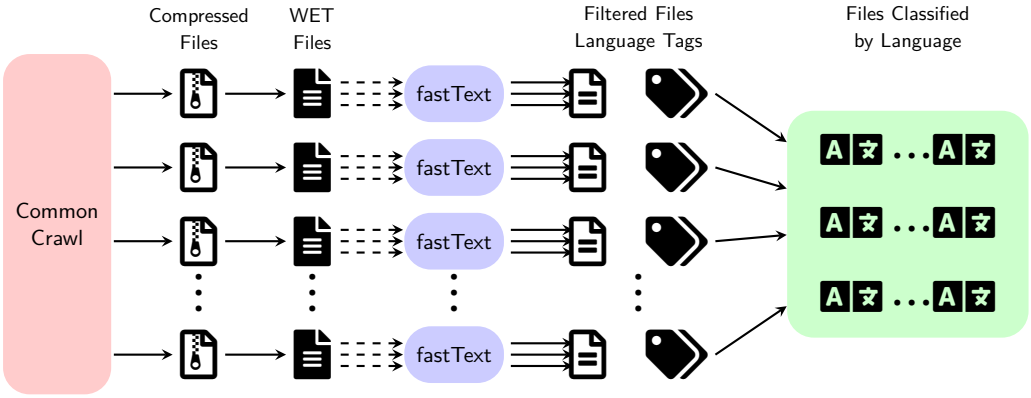


Figure 4.1: A scheme of the `goclassy` pipeline. The red square represents the Compressed WET files stored on Amazon Web Services. The icons represent the gzip files stored locally, the represents one of the 50K WET files. The represents the filtered file and the represents a file of language tags, one tag per line in . The represents one of the 166 classified files. Each arrow represents an asynchronous non-blocking worker and dotted arrows represent a line filtering process.

we launch a worker for each operation, and we bound the number of possible parallel operations at a given time by the number of available threads instead of the number of CPUs. We implement `goclassy` using the Go programming language[2] so we let the Go runtime[3] handle the scheduling of the processes. Thus, in our pipeline we don't have to wait for a whole WET file to download, decompress and classify in order to start downloading and processing the next one, a new file will start downloading and processing as soon as the scheduler is able to allocate a new process.

When using electromechanical mediums of storage, I/O blocking is one of the main problems one encounters. To overcome this, we introduced buffers in all our I/O operations, a feature that is not present in the fastText pre-processing pipeline. We also create, from the start, a file for each of the 176 languages that the pre-trained fastText language classifier is capable of recognizing, and we always leave them open, as we find that getting a file descriptor to each time we want to write, if we wanted to leave them open just when needed, introduces a big overhead.

We also do the filtering and cleaning processes at line level before feeding each line to the classifier, which makes us create a new filtered file so that we can have a correspondence with the tag file, which in turn will consume more space, but that will also reduce the amount of unnecessary classifications performed by fastText. The filtered and file tags are then read and lines are appended to its corresponding language file. The writing in the classification step is asynchronous, meaning that

[2] https://golang.org/
[3] https://golang.org/src/runtime/mprof.go

process writing a line to the filtered files does not wait for the classifier to write a tag on the tag file. Figure 4.1 shows the pipeline up to this point.

After all WET files are processed, we then use Isaac Whitfield's deduplication tool runiq[4] which is based on Yann Collet's xxhash64[5], an extremely fast non-cryptographic hash algorithm that is resistant to collisions. We finally use the Mark Adler's pigz[6] for data compression, as opposed to the canonical UNIX tools proposed in the original fastText pipeline. We add both tools to our concurrent pipeline, executing multiple instances of them in parallel, in order to ensure we use the most of our available resources at a given time.

Beyond improving the computational time required to classify this corpus, we propose a simple improvement on the cleaning scheme in the fastText pre-processing pipeline. This improvement allows our pipeline to better take into account the multilingual nature of Common Crawl; that is, we count UTF-8 characters instead of bytes for setting the lower admissible bound for the length of a line to be fed into the classifier. This straightforward modification on the fastText pre-processing pipeline assures we take into account the multiple languages present in Common Crawl that use non-ASCII encoded characters.

Given that our implementation is written in Go, we release binary distributions[7] of `goclassy` for all major operating systems. Both pigz and runiq are also available for all major operating systems.

4.2 Benchmarks

	10 files			100 files			200 files		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
real									
fastText	2m50s	6m45s	3m31s	13m46s	38m38s	17m39s	26m20s	47m48s	31m4s
`goclassy`	1m23s	3m12s	1m42s	7m42s	12m43s	9m8s	15m3s	15m47s	15m16s
user									
fastText	26m45s	27m2s	26m53s	4h21m	4h24m	4h23m	8h42m	8h48m	8h45m
`goclassy`	10m26s	12m53s	11m0s	1h46m	1h54m	1h49m	3h37m	3h40m	3h38m
sys									
fastText	40.14s	40.85s	40.56s	6m14s	6m17s	6m15s	12m26s	12m45s	12m31s
`goclassy`	37.34s	45.98s	39.67s	5m7s	5m34s	5m16s	9m57s	10m14s	10m5s

Table 4.1: Benchmarks are done using the UNIX `time` tool, are repeated 10 times each and are done for random samples of 10, 100 and 200 WET files. Only the classifying and filtering part are benchmarked. The table shows the minimum, maximum and mean time for the user, real and sys time over the 10 runs. Here "fastText" is used as short for the pipeline.

[4] https://github.com/whitfin/runiq
[5] https://github.com/Cyan4973/xxHash
[6] https://zlib.net/pigz/
[7] https://github.com/oscar-corpus/goclassy

We test both pipelines against one another in an infrastructure using traditional electromechanical storage mediums that are connected to the main processing machine via an Ethernet interface, that is, a low I/O speed environment as compared to an infrastructure where one would have an array of SSDs connected directly to the main processing machine via a high speed interface. We use a machine with an Intel® Xeon® Processor E5-2650 2.00 GHz, 20M Cache, and 203.1 GiB of RAM. We make sure that no other processes apart from the benchmark and the Linux system processes are run. We do not include downloading, decompression or deduplication in our benchmarks as downloading takes far too much time, and deduplication and compression were performed with third party tools that don't make part of our main contribution. We are mainly interested in seeing how the way the data is fed to the classifier impacts the overall processing time.

Benchmarks in table 4.1 of our `goclassy` pipeline show a drastic reduction in processing time compared to the original fastText prepossessing pipeline. We show that in our particular infrastructure, we are capable of reducing the *real* time as measured by the `time` UNIX tool almost always by half. The *user* time which represents the amount of CPU time spent in user-mode code (outside the kernel) within the process is almost three times lower for our `goclassy` pipeline, this particular benchmark strongly suggest a substantial reduction in energy consumption of `goclassy` with respect to the fastText pipeline.

As we understand that even an infrastructure with more than 20TB of free space in traditional electromechanical storage is not available to everyone and we propose a simple parametrization in our pipeline that actively deletes already processed data and that only downloads and decompresses files when needed, thus ensuring that no more than 10TB of storage are used at a given time. We nevertheless note that delaying decompression increases the amount of computation time, which is a trade-off that some users might make as it might be more suitable for their available infrastructure.

4.3 OSCAR 2019

We are aware that some users might not even have access to a big enough infrastructure to run our pipelines or just to store all the Common Crawl data. Moreover, even if previously used and cited in NLP and Machine Learning research, we note that at the time of OSCAR's 2019 publication there was no public distribution of Common Crawl that was filtered, classified by language and ready to use for Machine Learning or NLP applications. We thus decide to publish a pre-processed version of the November 2018 dump of Common Crawl which comprises usable data in 166 different languages, we publish[8] our version under the name OSCAR 2019 which is short for *Open Super-large Crawled Aggregated coRpus* 2019.

[8] `https://oscar-corpus.com/post/oscar-2019/` and `https://huggingface.co/datasets/oscar`

After processing all the data with `goclassy`, the size of the whole Common Crawl corpus is reduced to 6.3TB, but in spite of this considerable reduction, OSCAR 2019 still dwarfed all previously freely available corpora having more 800 billion "words" or space-separated tokens and noting that this in fact is an understatement of how big OSCAR 2019 really is, as some of the largest subcorpora within OSCAR 2019 such as Chinese and Japanese do not use spaces. The sizes in bytes for both the original and the deduplicated versions of OSCAR 2019 can be found in table A.1. OSCAR 2019 is published in both in *unshuffled* and *shuffled* distributions:

- The *unshuffled* distribution loosely respects the original documents, this is because by design `goclassy` considers that a *document* is a set of contiguous lines (i.e. coming from the same URL record) that share a language classification. Thus, if a URL record contains texts in multiple languages, `goclassy` will split this record in multiple documents. The *documents* here are separated by newlines. This *unshuffled* OSCAR 2019 is distributed from France under a research-only license, or from the USA through the *Hugging Face*'s datasets library under the *Creative Commons CC0 license* (*"no rights reserved"*).[9] This is in part due to the difference in copyright laws between the US and the EU.

- The *shuffled* distribution takes each language subcorpus of the *unshuffled* distribution of OSCAR 2019 and shuffles it at line level. There is no concept of document in this distribution of OSCAR 2019. As the original content is not reconstructive, we distribute the shuffled OSCAR 2019 from France under the *Creative Commons CC0 license* (*"no rights reserved"*).

4.4 Conclusions

We have presented `goclassy` a very efficient and concurrent pipeline for language classification and data cleaning and pre-processing, we have also presented OSCAR 2019 a substantially big Common Crawl-based corpus aimed at NLP application needing large quantities of raw textual data such as the pre-training of state-of-the-art language models. As we will see in further chapters, OSCAR 2019 would end up substantially increasing the amount of freely available data for medium to low resource languages, thus improving and facilitating NLP research for them. Moreover, our `goclassy` pipeline will continue to evolve and be fully rewritten into a new and optimized pipeline, greatly facilitating the production of large scale multilingual corpora in constrained pr low budget infrastructures. However, as OSCAR 2019 is still an automatically web-crawled corpus that at this point hadn't been manually audited, many questions remained about the quality of the data, at this point we didn't even know if producing a usable language model out of it was

[9] http://creativecommons.org/publicdomain/zero/1.0/

possible. These and other question will be discussed and answered in the following chapters.

5 A First Evaluation of the OSCAR Corpus

> In which we present the work of Ortiz Suárez et al. (2020b), who propose the first evaluation of OSCAR 2019 as a pre-training corpus for language modeling. This evaluation was done by selecting OSCAR subcorpora for 5 morphologically and tipologically different mid-resource languages and pre-training monolingual ELMo models (Peters et al., 2018) for each of them. These ELMo models are then attached to the UDPipe 2.0 architecture (Straka, 2018; Straka et al., 2019) and evaluated in dependency parsing and POS tagging.[1]

Having released OSCAR 2019, the first thing that we wanted to do with it was to evaluate how good it actually was for what it was mainly intended, that is, the pre-training of contextualized word embeddings that had just become available at the time we started working on OSCAR 2019. Such models included ULMFiT (Howard and Ruder, 2018), ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) among others at that time. For this first evaluation we decided to train ELMo contextualized word embeddings for 5 languages: Bulgarian, Catalan, Danish, Finnish and Indonesian. We train one set of embeddings using only Wikipedia data, and another set using only OSCAR 2019 data. We chose these languages primarily because they are morphologically and typologically different from one another, but also because all the OSCAR 2019 subcorpora for these languages were of a sufficiently manageable size such that the ELMo pre-training was doable in less than one month with the computational resources we had access to at the time. Contrary to *HIT-SCIR* team (Che et al., 2018), we do not impose any cap on the amount of data, and instead use the entirety of Wikipedia or OSCAR 2019 for each of our 5 chosen languages.

5.1 Corpora

Wikipedia is the biggest online multilingual open encyclopedia, comprising more than 40 million articles in 301 different languages. Because articles are curated by

[1] Contributions: I did the concrete work and designed most of it under the supervision of my coauthors and Ph.D. supervisors.

Language	Size	#Ktokens	#Kwords	#Ksentences
Bulgarian	609M	64,190	54,748	3,685
Catalan	1.1G	211,627	179,108	8,293
Danish	338M	60,644	52,538	3,226
Finnish	669M	89,580	76,035	6,847
Indonesian	488M	80,809	68,955	4,298

Table 5.1: Size of Wikipedia corpora, measured in bytes, thousands of tokens, words and sentences.

Language	Size	#Ktokens	#Kwords	#Ksentences
Bulgarian	14G	1,466,051	1,268,115	82,532
Catalan	4.3G	831,039	729,333	31,732
Danish	9.7G	1,828,881	1,620,091	99,766
Finnish	14G	1,854,440	1,597,856	142,215
Indonesian	16G	2,701,627	2,394,958	140,138

Table 5.2: Size of OSCAR 2019 subcorpora, measured in bytes, thousands of tokens, words and sentences.

language and written in an open collaboration model, its text tends to be of very high-quality in comparison to other free online resources. This is why Wikipedia has been extensively used in various NLP applications (Wu and Weld, 2010; Mihalcea, 2007; Al-Rfou' et al., 2013; Bojanowski et al., 2017). We downloaded the XML Wikipedia dumps[2] and extracted the plain-text from them using the `wikiextractor.py` script[3] from Giuseppe Attardi. We present the number of words and tokens available for each of our 5 languages in Table 5.1. Furthermore, we decided against deduplicating the Wikipedia data as the corpora are already quite small. We tokenize the 5 corpora using *UDPipe* (Straka and Straková, 2017).

As we did for Wikipedia, we tokenize OSCAR 2019 subcorpora for the 5 languages we chose for our study using UDPipe. Table 5.2 provides quantitative information about the 5 resulting tokenized corpora.

5.1.1 Noisiness

We wanted to address (Trinh and Le, 2018) and (Radford et al., 2019)'s criticisms of Common Crawl, so we devised a simple method to measure how noisy the OSCAR 2019 subcorpora were for our 5 languages. We randomly extract a number of lines from each corpus, such that the resulting random sample contains one

[2]XML dumps from April 4, 2019.
[3]Available here.

Language	OOV Wikipedia	OOV OSCAR 2019
Bulgarian	60,879	66,558
Catalan	34,919	79,678
Danish	134,677	123,299
Finnish	266,450	267,525
Indonesian	116,714	124,607

Table 5.3: Number of out-of-vocabulary words in random samples of 1M words for OSCAR 2019 and Wikipedia.

million words.[4] Likewise, we test if the words are in the corresponding *GNU Aspell*[5] dictionary. We repeat this task for each of the 5 languages, for both the OSCAR and the Wikipedia corpora. We compile in Table 5.3 the number of out-of-vocabulary tokens for each corpus.

As expected, this simple metric shows that in general the OSCAR samples contain more out-of-vocabulary words than the Wikipedia ones. However, the difference in magnitude between the two is strikingly lower than one would have expected in view of the criticisms by Trinh and Le (2018) and Radford et al. (2019), thereby validating the usability of Common Crawl data when it is properly filtered, as was achieved by in the OSCAR 2019 corpus. We even observe that, for Danish, the number of out-of-vocabulary words in OSCAR is lower than that on Wikipedia.

5.2 Experimental Setting

The main goal of this paper is to show the impact of training data on contextualized word representations when applied in particular downstream tasks. To this end, we train different versions of the *Embeddings from Language Models* (ELMo) (Peters et al., 2018) for both the Wikipedia and OSCAR 2019 corpora, for each of our selected 5 languages. We save the models' weights at different number of epochs for each language, in order to test how corpus size affect the embeddings and to see whether and when overfitting happens when training ELMo on smaller corpora.

We take each of the trained ELMo models and use them in conjunction with the UDPipe 2.0 (Straka, 2018; Straka et al., 2019) architecture for dependency parsing and POS-tagging to test our models. Furthermore, we train UDPipe 2.0 using gold tokenization and segmentation for each of our ELMo models, the only thing that changes from training to training is the ELMo model as hyperparameters always remain at the default values (except for number of training tokens) (Peters et al., 2018).

[4]We remove tokens that are capitalized or contain less than 4 UTF-8 encoded characters, allowing us to remove bias against Wikipedia, which traditionally contains a large quantity of proper nouns and acronyms.

[5]http://aspell.net/

5.2.1 Contextualized word embeddings

Embeddings from Language Models (ELMo) (Peters et al., 2018) is an LSTM-based language model. More precisely, it uses a bidirectional language model, which combines a forward and a backward LSTM-based language model. ELMo also computes a context-independent token representation via a CNN over characters.

We train ELMo models for Bulgarian, Catalan, Danish, Finnish and Indonesian using the OSCAR 2019 subcorpora on the one hand and the Wikipedia corpora on the other. We train each model for 10 epochs, as was done for the original English ELMo (Peters et al., 2018). Likewise, we save checkpoints at 1st, 3rd and 5th epoch in order to investigate some concerns about possible overfitting for smaller corpora (Wikipedia in this case) raised by the original ELMo authors.[6]

5.2.2 UDPipe 2.0

For our POS tagging and dependency parsing evaluation, we use UDPipe 2.0, which has a freely available and ready to use implementation.[7] This architecture was submitted as a participant to the *2018 CoNLL Shared Task* (Zeman et al., 2018), obtaining the 3rd place in LAS ranking. UDPipe 2.0 is a multi-task model that predicts POS tags, lemmas and dependency trees jointly.

The original UDPipe 2.0 implementation calculates 3 different embeddings, namely:

- *Pre-trained word embeddings*: In the original implementation, the Wikipedia version of fastText embeddings is used (Bojanowski et al., 2017); we replace them in favor of the newer Common-Crawl-based fastText embeddings trained by Grave et al. (2018).
- *Trained word embeddings*: Randomly initialized word representations that are trained with the rest of the network.
- *Character-level word embeddings*: Computed using bi-directional GRUs of dimension 256. They represent every UTF-8 encoded character with two 256 dimensional vectors, one for the forward and one for the backward layer. These two vector representations are concatenated and are trained along the whole network.

After the CoNLL 2018 Shared Task, the UDPipe 2.0 authors added the option to concatenate contextualized representations to the embedding section of the network (Straka et al., 2019), we use this new implementation, and we concatenate our pre-trained deep contextualized ELMo embeddings to the three embeddings mentioned above.

[6] https://github.com/allenai/bilm-tf/issues/135
[7] https://github.com/CoNLL-UD-2018/UDPipe-Future

Treebank	#Ktokens	#Ksentences
Bulgarian-BTB	156	11
Catalan-AnCora	530	17
Danish-DDT	100	6
Finnish-FTB	159	19
Finnish-TDT	202	15
Indonesian-GSD	121	6

Table 5.4: Size of treebanks, measured in thousands of tokens and sentences.

Once the embedding step is completed, the concatenation of all vector representations for a word are fed to two shared bidirectional LSTM (Hochreiter and Schmidhuber, 1997) layers. The output of these two BiLSTMS is then fed to two separate specific LSTMs:

- The tagger- and lemmatizer-specific bidirectional LSTMs, with Softmax classifiers on top, which process its output and generate UPOS, XPOS, UFeats and Lemmas. The lemma classifier also takes the character-level word embeddings as input.
- The parser-specific bidirectional LSTM layer, whose output is then passed to a bi-affine attention layer (Dozat and Manning, 2017) producing labeled dependency trees.

5.2.3 Treebanks

To train the selected parser and tagger (cf. Section 5.2.2) and evaluate the pre-trained language models in our 5 languages, we run our experiments using the Universal Dependencies (UD)[8] paradigm and its corresponding UD POS tag set (Petrov et al., 2012). We use all the treebanks available for our five languages in the UD treebank collection version 2.2 (Nivre et al., 2018), which was used for the CoNLL 2018 shared task, thus we perform our evaluation tasks in 6 different treebanks (see Table 5.4 for treebank size information).

- *Bulgarian BTB*: Created at the Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, it consists of legal documents, news articles and fiction pieces.
- *Catalan-AnCora*: Built on top of the Spanish-Catalan *AnCora corpus* (Taulé et al., 2008), it contains mainly news articles.
- *Danish-DDT*: Converted from the *Danish Dependency Treebank* (Buch-Kromann, 2003). It includes news articles, fiction and non fiction texts and oral transcriptions.

[8] https://universaldependencies.org

- *Finnish-FTB*: Consists of manually annotated grammatical examples from VISK[9] (The Web Version of the Large Grammar of Finnish).
- *Finnish-TDT*: Based on the Turku Dependency Treebank (TDT). Contains texts from Wikipedia, Wikinews, news articles, blog entries, magazine articles, grammar examples, Europarl speeches, legal texts and fiction.
- *Indonesian-GSD*: Includes mainly blog entries and news articles.

5.3 Results & Discussion

Treebank	Model	UPOS	UAS	LAS
Bulgarian BTB	UDify	98.89	95.54	92.40
	UDPipe 2.0	98.98	93.38	90.35
	+mBERT	99.20	95.34	92.62
	+ELMo$_{\text{Wikipedia}}$	99.17	94.93	92.05
	+ELMo$_{\text{OSCAR}}$	**99.40**	**96.01**	**93.56**
Catalan-AnCora	UDify	98.89	94.25	92.33
	UDPipe 2.0	98.88	93.22	91.06
	+mBERT	**99.06**	**94.49**	92.74
	+ELMo$_{\text{Wikipedia}}$	99.05	93.99	92.24
	+ELMo$_{\text{OSCAR}}$	**99.06**	**94.49**	**92.88**
Danish-DDT	UDify	97.50	87.76	84.50
	UDPipe 2.0	97.78	86.88	84.31
	+mBERT	98.21	89.32	87.24
	+ELMo$_{\text{Wikipedia}}$	98.45	89.05	86.92
	+ELMo$_{\text{OSCAR}}$	**98.62**	**89.84**	**87.95**

Treebank	Model	UPOS	UAS	LAS
Finnish-FTB	UDify	93.80	86.37	81.40
	UDPipe 2.0	96.65	90.68	87.89
	+mBERT	96.97	91.68	89.02
	+ELMo$_{\text{Wikipedia}}$	97.27	92.05	89.62
	+ELMo$_{\text{OSCAR}}$	**98.13**	**93.81**	**92.02**
Finnish-TDT	UDify	94.43	86.42	82.03
	UDPipe 2.0	97.45	89.88	87.46
	+mBERT	97.57	91.66	89.49
	+ELMo$_{\text{Wikipedia}}$	97.65	91.60	89.34
	+ELMo$_{\text{OSCAR}}$	**98.36**	**93.54**	**91.77**
Indonesian-GSD	UDify	93.36	86.45	80.10
	UDPipe 2.0	93.69	85.31	78.99
	+mBERT	94.09	86.47	80.40
	+ELMo$_{\text{Wikipedia}}$	93.94	86.16	80.10
	+ELMo$_{\text{OSCAR}}$	**94.12**	**86.49**	**80.59**

Table 5.5: Scores from UDPipe 2.0 (from Kondratyuk and Straka, 2019), the previous state-of-the-art models UDPipe 2.0+mBERT (Straka et al., 2019) and UDify (Kondratyuk and Straka, 2019), and our ELMo-enhanced UDPipe 2.0 models. Test scores are given for UPOS, UAS and LAS in all five languages. Best scores are shown in bold, second-best scores are underlined.

5.3.1 Parsing and POS tagging results

We use UDPipe 2.0 without contextualized embeddings as our baseline for POS tagging and dependency parsing. However, we did not train the model without contextualized word embedding ourselves. We instead take the scores as they are reported in (Kondratyuk and Straka, 2019). We also compare our UDPipe 2.0 + ELMo models against the state-of-the-art results (assuming gold tokenization) for these languages, which are either UDify (Kondratyuk and Straka, 2019) or UDPipe 2.0 + mBERT (Straka et al., 2019).

Results for UPOS, UAS and LAS are shown in Table 5.5. We obtain the state of the art for the three metrics in each of the languages with the UDPipe 2.0 + ELMo$_{\text{OSCAR}}$

[9] http://scripta.kotus.fi/visk

models. We also see that in every single case the UDPipe 2.0 + $\text{ELMo}_{\text{OSCAR}}$ result surpasses the UDPipe 2.0 + $\text{ELMo}_{\text{Wikipedia}}$ one, suggesting that the size of the pre-training data plays an important role in downstream task results. This also supports our hypothesis that the OSCAR corpus, being multi-domain, exhibits a better coverage of the different styles, genres and uses present at least in these 5 languages.

Taking a closer look at the results for Danish, we see that $\text{ELMo}_{\text{Wikipedia}}$, which was trained with a mere 300 MB corpus, does not show any sign of overfitting, as the UDPipe 2.0 + $\text{ELMo}_{\text{Wikipedia}}$ results considerably improve the UDPipe 2.0 baseline. This is the case for all of our $\text{ELMo}_{\text{Wikipedia}}$ models as we never see any evidence of a negative impact when we add them to the baseline model. In fact, the results of UDPipe 2.0 + $\text{ELMo}_{\text{Wikipedia}}$ give better than previous state-of-the-art results in all metrics for the Finnish-FTB and in UPOS for the Finnish-TDT. The results for Finnish are actually quite interesting, as mBERT was pre-trained on Wikipedia and here we see that the multilingual setting in which UDify was fine-tuned exhibits sub-baseline results for all metrics, and that the UDPipe + mBERT scores are often lower than those of our UDPipe 2.0 + $\text{ELMo}_{\text{Wikipedia}}$. This actually suggests that even though the multilingual approach of mBERT (in pre-training) or UDify (in pre-training and fine-tuning) leads to better performance for high-resource languages or languages that are closely related to high-resource languages, it might also significantly degrade the representations for more isolated or even simply more morphologically rich languages like Finnish. In contrast, our monolingual approach with UDPipe 2.0 + $\text{ELMo}_{\text{OSCAR}}$ improves the previous SOTA considerably, by more than 2 points for some metrics. Note however that Indonesian, which might also be seen as a relatively isolated language, does not behave in the same way as Finnish.

5.3.2 Impact of the number of training epochs

An important topic we wanted to address with our experiments was that of *overfitting* and the number of epochs one should train the contextualized embeddings for. The ELMo authors have expressed that increasing the number of training epochs is generally better, as they argue that training the ELMo model for longer reduces held-out perplexity and further improves downstream task performance.[10] This is why we intentionally fully pre-trained the $\text{ELMo}_{\text{Wikipedia}}$ to the 10 epochs of the original ELMo paper, as its authors also expressed concern over the possibility of overfitting for smaller corpora. We thus save checkpoints for each of our ELMo model at the 1, 3, 5 and 10 epoch marks so that we can properly probe for overfitting. The scores of all checkpoints are reported in Table 5.6. Here again we do not train the UDPipe 2.0 baselines without embedding, we just report the scores published in Kondratyuk and Straka (2019).

[10] Their comments on the matter can be found on https://github.com/allenai/bilm-tf/issues/135.

The first striking finding is that even though all our Wikipedia data sets are smaller than 1 GB in size (except for Catalan), none of the $\text{ELMo}_{\text{Wikipedia}}$ models show any sign of overfitting, as the results continue to improve for all metrics the more we train the ELMo models, with the best results consistently being those of the fully trained 10 epoch ELMos. For all of our Wikipedia models, but those of Catalan and Indonesian, we see sub-baseline results at 1 epoch; training the model for longer is better, even if the corpora are small.

Treebank	Model	UPOS	UAS	LAS
Bulgarian BTB	UDPipe 2.0	98.98	93.38	90.35
	+$\text{ELMo}_{\text{Wikipedia}(1)}$	98.81	93.60	90.21
	+$\text{ELMo}_{\text{Wikipedia}(3)}$	99.01	94.32	91.36
	+$\text{ELMo}_{\text{Wikipedia}(5)}$	99.03	94.32	91.38
	+$\text{ELMo}_{\text{Wikipedia}(10)}$	99.17	94.93	92.05
	+$\text{ELMo}_{\text{OSCAR}(1)}$	99.28	95.45	92.98
	+$\text{ELMo}_{\text{OSCAR}(3)}$	99.34	95.58	93.12
	+$\text{ELMo}_{\text{OSCAR}(5)}$	99.34	95.63	93.25
	+$\text{ELMo}_{\text{OSCAR}(10)}$	**99.40**	**96.01**	**93.56**
Catalan-AnCora	UDPipe 2.0	98.88	93.22	91.06
	+$\text{ELMo}_{\text{Wikipedia}(1)}$	98.93	93.24	91.21
	+$\text{ELMo}_{\text{Wikipedia}(3)}$	99.02	93.75	91.93
	+$\text{ELMo}_{\text{Wikipedia}(5)}$	99.04	93.86	92.05
	+$\text{ELMo}_{\text{Wikipedia}(10)}$	99.05	93.99	92.24
	+$\text{ELMo}_{\text{OSCAR}(1)}$	99.07	93.92	92.29
	+$\text{ELMo}_{\text{OSCAR}(3)}$	**99.10**	94.29	92.69
	+$\text{ELMo}_{\text{OSCAR}(5)}$	99.07	94.38	92.75
	+$\text{ELMo}_{\text{OSCAR}(10)}$	99.06	**94.49**	**92.88**
Danish-DDT	UDPipe 2.0	97.78	86.88	84.31
	+$\text{ELMo}_{\text{Wikipedia}(1)}$	97.47	86.98	84.15
	+$\text{ELMo}_{\text{Wikipedia}(3)}$	98.03	88.16	85.81
	+$\text{ELMo}_{\text{Wikipedia}(5)}$	98.15	88.24	85.96
	+$\text{ELMo}_{\text{Wikipedia}(10)}$	98.45	89.05	86.92
	+$\text{ELMo}_{\text{OSCAR}(1)}$	98.50	89.47	87.43
	+$\text{ELMo}_{\text{OSCAR}(3)}$	98.59	89.68	87.77
	+$\text{ELMo}_{\text{OSCAR}(5)}$	98.59	89.46	87.64
	+$\text{ELMo}_{\text{OSCAR}(10)}$	**98.62**	**89.84**	**87.95**
Finnish-FTB	UDPipe 2.0	96.65	90.68	87.89
	+$\text{ELMo}_{\text{Wikipedia}(1)}$	95.86	89.63	86.39
	+$\text{ELMo}_{\text{Wikipedia}(3)}$	96.76	91.02	88.27
	+$\text{ELMo}_{\text{Wikipedia}(5)}$	96.97	91.66	89.04
	+$\text{ELMo}_{\text{Wikipedia}(10)}$	97.27	92.05	89.62
	+$\text{ELMo}_{\text{OSCAR}(1)}$	97.91	93.41	91.43
	+$\text{ELMo}_{\text{OSCAR}(3)}$	98.00	**93.99**	91.98
	+$\text{ELMo}_{\text{OSCAR}(5)}$	**98.15**	93.98	**92.24**
	+$\text{ELMo}_{\text{OSCAR}(10)}$	98.13	93.81	92.02
Finnish-TDT	UDPipe 2.0	97.45	89.88	87.46
	+$\text{ELMo}_{\text{Wikipedia}(1)}$	96.73	89.11	86.33
	+$\text{ELMo}_{\text{Wikipedia}(3)}$	97.55	90.84	88.50
	+$\text{ELMo}_{\text{Wikipedia}(5)}$	97.55	91.11	88.88
	+$\text{ELMo}_{\text{Wikipedia}(10)}$	97.65	91.60	89.34
	+$\text{ELMo}_{\text{OSCAR}(1)}$	98.27	93.03	91.29
	+$\text{ELMo}_{\text{OSCAR}(3)}$	98.38	**93.60**	**91.83**
	+$\text{ELMo}_{\text{OSCAR}(5)}$	**98.39**	93.57	91.80
	+$\text{ELMo}_{\text{OSCAR}(10)}$	98.36	93.54	91.77
Indonesian-GSD	UDPipe 2.0	93.69	85.31	78.99
	+$\text{ELMo}_{\text{Wikipedia}(1)}$	93.70	85.81	79.46
	+$\text{ELMo}_{\text{Wikipedia}(3)}$	93.90	86.04	79.72
	+$\text{ELMo}_{\text{Wikipedia}(5)}$	94.04	85.93	79.97
	+$\text{ELMo}_{\text{Wikipedia}(10)}$	93.94	86.16	80.10
	+$\text{ELMo}_{\text{OSCAR}(1)}$	93.95	86.25	80.23
	+$\text{ELMo}_{\text{OSCAR}(3)}$	94.00	86.21	80.14
	+$\text{ELMo}_{\text{OSCAR}(5)}$	**94.23**	86.37	80.40
	+$\text{ELMo}_{\text{OSCAR}(10)}$	94.12	**86.49**	**80.59**

Table 5.6: UPOS, UAS and LAS scores for the UDPipe 2.0 baseline reported by (Kondratyuk and Straka, 2019), plus the scores for checkpoints at 1, 3, 5 and 10 epochs for all the $\text{ELMo}_{\text{OSCAR}}$ and $\text{ELMo}_{\text{Wikipedia}}$. All scores are test scores. Best $\text{ELMo}_{\text{OSCAR}}$ scores are shown in bold while best $\text{ELMo}_{\text{Wikipedia}}$ scores are underlined.

$\text{ELMo}_{\text{OSCAR}}$ models exhibit exactly the same behavior as $\text{ELMo}_{\text{Wikipedia}}$ models where the scores continue to improve the longer they are pre-trained, except for the case of Finnish. Here we actually see an unexpected behavior where the model performance caps around the 3$^{\text{rd}}$ to 5$^{\text{th}}$ epoch. This is surprising because the Finnish OSCAR 2019 subcorpus is more than 20 times bigger than our smallest Wikipedia corpus, the Danish Wikipedia, that did not exhibit this behavior. As previously mentioned Finnish is morphologically richer than the other languages in which we trained ELMo, we hypothesize that the representation space given by the ELMo embeddings might not be sufficiently big to extract more features from the Finnish OSCAR subcorpus beyond the 5$^{\text{th}}$ epoch mark, however in order to test this we would need to train a larger language model like BERT which is sadly beyond our computing infrastructure limits (cf. Appendix B.1). However, we do note that pre-

training our current language model architectures in a morphologically rich language like Finnish might actually better expose the limits of our existing approaches to language modeling.

One last thing that it is important to note with respect to the number of training epochs is that even though we fully pre-trained our $\text{ELMo}_{\text{Wikipedia}}$'s and $\text{ELMo}_{\text{OSCAR}}$'s to the recommended 10 epoch mark, and then compared them against one another, the number of training steps between both pre-trained models differs drastically due to the big difference in corpus size (for Indonesian, for instance, 10 epochs correspond to 78K steps for $\text{ELMo}_{\text{Wikipedia}}$ and to 2.6M steps for OSCAR; the complete picture is provided in the Appendix, in Table B.2). In fact, we can see in Table 5.6 that all the UDPipe 2.0 + $\text{ELMo}_{\text{OSCAR(1)}}$ perform better than the UDPipe 2.0 + $\text{ELMo}_{\text{Wikipedia(1)}}$ models across all metrics. Thus, we believe that talking in terms of training steps as opposed to training epochs might be a more transparent way of comparing two pre-trained models.

5.4 Conclusions

In this chapter, we have explored the use of the Common-Crawl-based OSCAR 2019 corpora to train ELMo contextualized embeddings for five typologically diverse mid-resource languages. We have compared them with Wikipedia-based ELMo embeddings on two classical NLP tasks, POS tagging and parsing, using state-of-the-art neural architectures at the end of 2019. Our goal was to explore whether the noisiness level of Common Crawl data, often invoked to criticize the use of such data, could be compensated by its larger size; for some languages, the OSCAR 2019 corpus is several orders of magnitude larger than the corresponding Wikipedia. Firstly, we found that when properly filtered, Common Crawl data is not massively noisier than Wikipedia. Secondly, we show that embeddings trained using OSCAR 2019 data consistently outperform Wikipedia-based embeddings, to the extent that they allow us to improve the state of the art in POS tagging and dependency parsing for all the 6 chosen treebanks. Thirdly, we observe that more training epochs generally results in better embeddings even when the training data is relatively small, as is the case for Wikipedia.

Our experiments show that Common-Crawl-based data such as the OSCAR corpus can be used to train high-quality contextualized embeddings, even for languages for which more standard textual resources lack volume or genre variety. This could result in better performances in a number of NLP tasks for many non highly resourced languages. However, we are aware that this first evaluation of the OSCAR 2019 remains limited both in terms of methodology and in terms of the actual portion of the whole corpus that was evaluated. Automated evaluations like this one will not give us a complete assessment of the quality of the corpus beyond its usefulness for the training of contextualized embeddings. This is why in the following chapter we

will discuss a more thorough and extensive audit of the OSCAR 2019 corpus as well as other web-crawled corpora, which was the result of an international collaboration with a diverse team of more than 50 researchers.

6 Quality at a Glance: An Audit of OSCAR 2019 and other Web-Crawled Datasets

In which we present the work of Kreutzer et al. (2022), who propose the first manual audit of OSCAR 2019 along other 4 crawled corpora. For the audit 51 volunteers from the NLP community were recruited, covering about 70 languages with proficient language skills. The study proposes solutions for effective, low-effort data auditing, including an error taxonomy. The study reflects on the potential harm of low-quality data releases for low-resource languages, and provides a set of recommendations for future multilingual data releases.[1]

Having done a first automatic evaluation of a small portion of the OSCAR 2019 corpus for a selection for 5 mid-resource languages, we wanted to better assess the global quality of the corpus specially for low-resource languages. To accomplish this, we participated in a collaborative effort to manually audit OSCAR 2019 and other 4 crawled corpora that have been extensively used in NLP research in the last few years.

Thus, to shed light on the quality of data crawls, specially for the lowest resource languages, we perform a manual data audit for 230 per-language subsets of five major crawled multilingual datasets:[2] CCAligned (El-Kishky et al., 2020), ParaCrawl (Esplà et al., 2019; Bañón et al., 2020), WikiMatrix (Schwenk et al., 2021), OSCAR 2019 (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020b) and mC4 (Xue et al., 2021). We propose solutions for effective, low-effort data auditing (Section 6.1), including an error taxonomy. Our quantitative analysis reveals surprisingly low amounts of valid in-language data, and identifies systematic issues across datasets and languages. In addition, we find that a large number of datasets is labeled with nontransparent or incorrect language codes (Section 6.2). This leads us to reflect on the potential harm

[1] Contributions: I annotated 5 subcorpora for OSCAR 2019, and 2 subcorpora for ParaCrawl 7.1. I also facilitated the access to the OSCAR samples, helped with statistics about the OSCAR corpus specially when they involved operations over the entire corpus. Finally, I actively participated in the writing of the scientific article.

[2] Annotations are available for download (last accessed: 12 Oct 2021).

of low-quality data releases for low-resource languages (Section 6.3), and provide a set of recommendations for future multilingual data releases (Section 6.4).

6.1 Auditing Data Quality

None of the five selected datasets has been evaluated for quality on the sentence level (exception: several languages in ParaCrawl v3), and downstream evaluations are centered around a small fraction of higher-resource languages. This is insufficient for drawing conclusions about the quality of individual or aligned sentences (in parallel datasets), and about the entirety of languages. In addition, there might be a publication bias preventing negative results with any of the above corpora with lower quality being published.

To close this gap, we conduct a human data quality audit focused on the lowest-resource and most under-evaluated languages, but also covering mid- and high-resource languages for comparison.

6.1.1 Auditing Process

Participants We recruited 51 volunteers from the NLP community, covering about 70 languages with proficient language skills.[3] Each sentence is annotated by one rater. To verify our hypothesis that those annotations can be largely done by non-native speakers, we repeat a set of language expert annotations by a non-expert, and measure the accuracy of the non-expert.

Sample selection For each language in each dataset, we took a random sample of 100 lines, which may be anywhere from single words to short paragraphs depending on segmentation. We manually annotated them according to the error taxonomy described below. For WikiMatrix and CCAligned, we selected those languages that are paired with English, and for ParaCrawl, we also included those paired with Spanish ("total" counts in Table 6.2). We did not annotate all languages, but focused on the ones with the least number of sentences in each dataset (at least the smallest 10) and languages for which we found proficient speakers. Since we annotate the same maximum number of sentences[4] across all chosen languages regardless of their total number of sentences, the annotated samples are not an unbiased sample from the whole dataset.

[3] This surprisingly high number comes in part because there are many closely related languages, e.g. one person may be proficient enough to rate many different Slavic or Turkic languages even if only one is their native language.

[4] Some languages had fewer than 100 sentences.

Non-expert labeling strategies Although many of the volunteers were familiar with the languages in question or spoke related languages, in cases where no speaker of a relevant language could be found, volunteers used dictionaries and internet search to form educated guesses. We discuss this deeper in Appendix C.3 to highlight how much of this low-resource focused evaluation can actually be done by non-proficient speakers with relatively low effort. In general, we aim to find an upper bound on quality, so we encouraged annotators to be forgiving of translation mistakes when the overall meaning of the sentence or large parts thereof are conveyed, or when most of the sentence is in the correct language.

Effort The individual effort was dependent on the quality and complexity of the data, and on the annotator's knowledge of the language(s), e.g., it took from less than two minutes for an English native speaker to pass through 100 well-formed English sentences (or similarly to annotate languages with 0% in-language sentences), to two hours of "detective work" for well-formed content in languages for an annotator without familiarity.

Correct Codes	
`C`: *Correct translation, any*	Combined label for `CC`, `CB`, `CS`
`CC`: *Correct translation, natural sentence*	
`en` The Constitution of South Africa	`nso` Molaotheo wa Rephabliki ya Afrika Borwa
`en` Transforming your swimming pool into a pond	`de` Umbau Ihres Swimmingpools zum Teich
`CB`: *Correct translation, Boilerplate or low quality*	
`en` Reference number: 13634	`ln` Motango ya référence: 13634
`en` Latest Smell Stop Articles	`fil` Pinakabagong mga Artikulo Smell Stop
`CS`: *Correct translation, Short*	
`en` movies, dad	`it` cinema, papà
`en` Halloween - without me	`ay` Hallowen – janiw nayampejj
Error Codes	
`X`: *Incorrect translation, but both correct languages*	
`en` A map of the arrondissements of Paris	`kg` Paris kele mbanza ya kimfumu ya Fwalansa.
`en` Ask a question	`tr` Soru sor Kullanıma göre seçim
`WL`: *Source OR target wrong language, but both still linguistic content*	
`en` The ISO3 language code is zho	`zza` Táim eadra bracach mar bhionns na frogannaidhe.
`en` Der Werwolf — sprach der gute Mann,	`de` des Weswolfs, Genitiv sodann,
`NL`: *Not a language: at least one of source and target are not linguistic content*	
`en` EntryScan 4 _	`tn` TSA PM704 _
`en` organic peanut butter	`ckb` ������

Table 6.1: Annotation codes for parallel data with sentence pair examples. The language code before each sentence indicates the language it is supposed to be in.

Taxonomy In order to quantify errors, we developed a simple error taxonomy. Sentences and sentence pairs were annotated according to a simple rubric with error classes of Incorrect Translation (`X`, excluded for monolingual data), Wrong Language

(WL), and Non-Linguistic Content (NL). Of correct sentences (C), we further mark single words or phrases (CS) and boilerplate contents (CB). In addition, we asked annotators to flag offensive or pornographic content. Table 6.1 provides examples for parallel data, and Appendix C.2 contains detailed annotation instructions.

		Parallel			Monolingual	
		CCAligned	**ParaCrawl v7.1**	**WikiMatrix**	**OSCAR**	**mC4**
	#langs audited / total	65 / 119	21 / 38	20 / 78	51 / 166	48 / 108
	%langs audited	54.62%	55.26%	25.64%	30.72%	44.44%
	#sents audited / total	8037 / 907M	2214 / 521M	1997 / 95M	3517 / 8.4B	5314 / 8.5B
	%sents audited	0.00089%	0.00043%	0.00211%	0.00004%	0.00006%
macro	C	29.25%	76.14%	23.74%	87.21%	72.40%
	X	29.46%	19.17%	68.18%	-	-
	WL	9.44%	3.43%	6.08%	6.26%	15.98%
	NL	31.42%	1.13%	1.60%	6.54%	11.40%
	offensive	0.01%	0.00%	0.00%	0.14%	0.06%
	porn	5.30%	0.63%	0.00%	0.48%	0.36%
micro	C	53.52%	83.00%	50.58%	98.72%	92.66%
	X	32.25%	15.27%	47.10%	-	-
	WL	3.60%	1.04%	1.35%	0.52%	2.33%
	NL	10.53%	0.69%	0.94%	0.75%	5.01%
	offensive	0.00%	0.00%	0.00%	0.18%	0.03%
	porn	2.86%	0.33%	0.00%	1.63%	0.08%
	#langs =0% C	7	0	1	7	0
	#langs <50% C	44	4	19	11	9
	#langs >50% NL	13	0	0	7	1
	#langs >50% WL	1	0	0	3	4

Table 6.2: Averages of sentence-level annotations across datasets and selected languages. Macro-avg: Each language is weighted equally in the aggregation, regardless of its size. Micro-avg: Each label is weighted by the fraction of sentences for that language in the overall annotated corpus, i.e., the annotations for higher-represented languages are upweighted, and annotations for lower-represented languages are downweighted. The bottom rows contain the number of languages that have 0% labeled C etc. Note that these are not true expectations since the languages audited were not randomly sampled.

6.1.2 Human Audit Results

Interpretation of Results For each language, we compute the percentage of each label within the 100 audited sentences. Then, we either aggregate the labels across languages with equal weights (macro-average), or weight them according to their presence in the overall dataset (micro-average). Results are shown in Table 6.2. The statistics for the correct codes (CC, CB, CS) are combined as C. The number of languages, the numbers of sentences per language and the choice of languages differ across datasets, both in the original release and in the selection for our audit, so the

comparison of numbers across datasets has to be taken with a grain of salt. Since the numbers are based on a small sample of sentences that were partially annotated by non-experts, the error statistics are only rough estimates. Our audit captures a decent ratio of languages (25–55%, 2nd row in Table 6.2), but only a tiny fraction of the overall number of sentences (0.00004–0.002%). When we speak of "low-" and "high"-resource languages, we mean languages with smaller or larger representation in the datasets at hand. When reporting language-specific results we use the original language identifiers of the datasets.

Which datasets have quality issues? The macro-averaged results show that the ratio of correct samples (`C`) ranges from 24% to 87%, with a large variance across the five audited datasets. Particularly severe problems were found in CCAligned and WikiMatrix, with 44 of the 65 languages that we audited for CCAligned containing under 50% correct sentences, and 19 of the 20 in WikiMatrix. In total, 15 of the 205 language specific samples (7.3%) contained not a single correct sentence. For the parallel datasets we are also interested in the quantity of misaligned/mistranslated sentences (`X`). For WikiMatrix, two-thirds of the audited samples were on average misaligned. We noticed that sentences were often similar in structure, but described different facts (see Table 6.5). This might originate from the nature of the underlying Wikipedia articles, since they are often comparable rather than parallel (Schwenk et al., 2021).

Figure 6.1 illustrates per-corpus correctness more completely, showing for each dataset what percent of audited corpora are under each possible threshold of correctness.

Why haven't these problems been reported before? The findings above are averaged on a per-language basis (i.e. macro-average), and therefore give low and high-resource languages equal weight. If we instead estimate the quality on a per-sentence basis, i.e. down-weight lower-resource languages in the computation of the average, the numbers paint a more optimistic picture ("micro" block in Table 6.2). This is especially relevant for the monolingual datasets because they contain audits for English, which makes up for 43% of all sentences in OSCAR 2019 and 36% in mC4. To illustrate the effect of this imbalance: A random sample from the entire mC4 dataset with over 63% chance will be from one of the 8 largest languages (`en`, `ru`, `es`, `de`, `fr`, `it`, `pt`, `pl`, >100M sentences each), of which all have near perfect quality. Analogously, evaluation and tuning of web mining pipelines and resulting corpora in downstream applications focused largely on higher-resource languages (Section 2.2.2), so the low quality of underrepresented languages might go unnoticed if there is no dedicated evaluation, or no proficient speakers are involved in the curation (Nekoto et al., 2020).

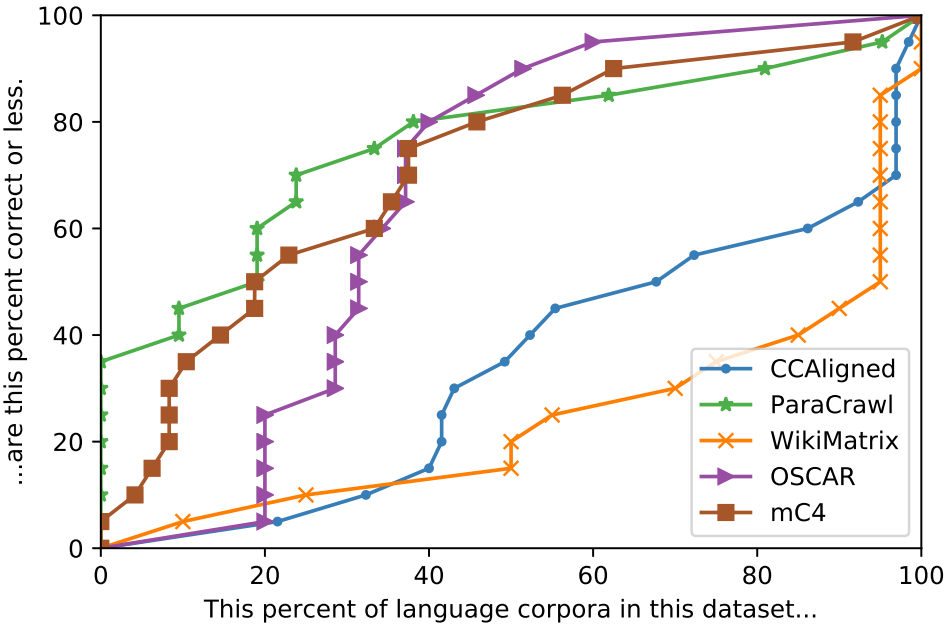
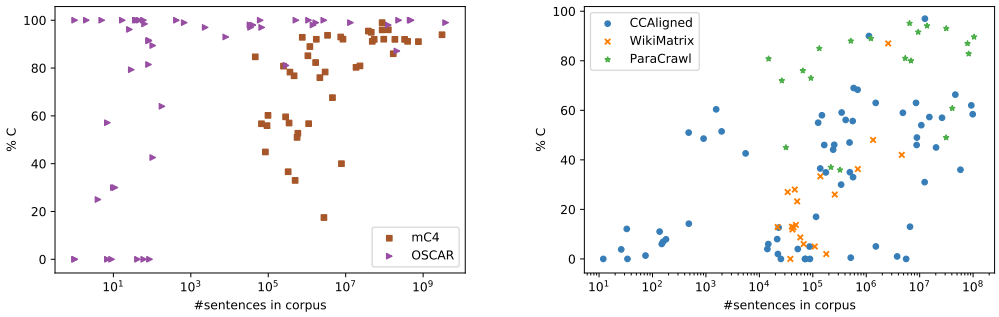


Figure 6.1: Fraction of languages in each dataset below a given quality threshold (percent correct).

How much content is nonlinguistic or in the wrong language? Nonlinguistic content is a more common problem than wrong-language content. Among the parallel datasets, CCAligned contains the highest percentage of nonlinguistic content, at 31.42% on average across all rated corpora, and also the highest percent of wrong-language content, at 9.44%. Among the monolingual datasets, mC4 contains the highest ratio both of sentences in incorrect languages (15.98% average) and nonlinguistic content (11.40% average), with 4 of the 48 audited languages having more than 50% contents in other languages. The low amount of wrong language in ParaCrawl shows the benefits of selecting domains by the amount in-language text, but the dataset also covers the smallest amount of languages. The low ratio of wrong language samples in OSCAR may reflect the success of line-level LangID filtering. These numbers provide evidence that more research in LangID could improve the overall quality, especially with respect to nonlinguistic content.

Which languages got confused? The languages that were confused were frequently related higher-resource languages. However, there were also a significant number of "out-of-model cousin" cases, where languages not supported by the LangID model ended up in a similar-seeming language. For instance in mC4, much of the Shona (`sn`, Bantu language spoken in Zimbabwe and Mozambique) corpus is actually

Kinyarwanda (rw, Bantu language spoken in mostly in Rwanda and Uganda)—and, peculiarly, much of the Hawaiian (haw, Polynesian language spoken in Hawaii) is actually Twi (tw/ak, Central Tano language spoken mostly in Ghana).



(a) Monolingual corpora (b) Parallel corpora

Figure 6.2: Percentage of sentences labeled as correct vs. log N sentences for all audited languages.

Do low-resource languages have lower quality? Low-resource datasets tend to have lower human-judged quality. The Spearman rank correlation between quality (%C) and size is positive in all cases. The trend is strongest for mC4 ($r = 0.66$), and gradually declines for CCAligned ($r = 0.53$), WikiMatrix ($r = 0.49$), ParaCrawl ($r = 0.43$), and OSCAR ($r = 0.37$). Figure 6.2 compares the number of sentences for each language against the proportion of correct sentences: Not all higher-resource languages ($> 10^6$ sentences) have high quality, in particular for CCAligned (e.g. Javanese (en-jv_ID) with 5%C, or Tagalog (en-tl_XX) with 13%C). For mid-resource languages (10^4–10^6 sentences) the picture is inconclusive, with some languages having high quality, and others having extremely low quality, even within the same datasets, e.g. Urdu in CCAligned en-ur_PK has 100%C vs. its romanized counterpart en-ur_PK_rom 0.5% C. For individual error codes trends are less clear (not depicted).

Which languages have the lowest quality? Across datasets we observe that the quality is particularly poor for languages that are included in romanized script (_rom/_latn), but are more commonly written in other scripts, e.g., Urdu (ur), Japanese (ja), Arabic (ar). These are not transliterations of other scripts, but mostly contain non-linguistic material or wrong languages (e.g. the romanized Japanese corpus in mC4 (ja_latn) contains Spanish, French, English, Portuguese, amongst others). In terms of geography, the poorest quality is found for African languages (Bambara (bm), Fula (ff), Kikongo (kg), Luganda (lg), Lingala (ln), Norther Sotho (nso), Oromo (om), Shona (sn), Somali (so), Tswana (tn), Wolof (wo)), minority languages in Europe and the Middle East that are closely related to higher-resource

languages (Azerbaijani (`az-IR`), North Frisian (`frr`), Neapolitan (`nap`), Silesian (`szl`), Zaza (`zza`)), lesser spoken Chinese languages sharing a script with Mandarin (Yue (`yue`), Wu (`wuu`)), four major Austronesian (Central Bikol (`bcl`), Chavacano (`cbk`), Javanese (`jv`), Sundanese (`su`)), and some South-Asian languages, in particular Sinhala (`si`). Appendix C.4 contains the detailed per-language statistics for all corpora.

What is the incidence of offensive and pornographic content? Overall, the sampled sentences did not contain a large amount of offensive contents. However, there were notable amounts of pornographic content (> 10%) found in CCAligned for 11 languages.

	es_XX	bm_ML	yo_NG	tr_TR	ku_TR	zh_CN	af_ZA	jv_ID	zh_TW	it_IT	**mean**
Acc-6	0.58	0.73	0.41	0.45	0.43	0.55	0.65	0.55	0.46	0.55	0.66
Acc-4	0.77	0.73	0.60	0.55	0.56	0.72	0.72	0.57	0.58	0.66	0.72
Acc-2	0.91	0.96	0.72	0.64	0.71	0.79	0.77	0.92	0.81	0.69	0.79

Table 6.3: Rater evaluation for a subset of audits from **CCAligned** (aligned with English) measured by the accuracy (Acc-n) of annotations by non-proficient speaker against annotations by proficient speakers.

	tyv	rm	bar	eml	zh	la	**mean**
Acc-6	1.0	0.98	1.0	1.0	0.86	1.0	0.98
Acc-4	1.0	1.0	1.0	1.0	0.87	1.0	0.98
Acc-2	1.0	1.0	1.0	1.0	0.87	1.0	0.98

Table 6.4: Rater evaluation for a subset of audits from **OSCAR 2019** measured by the accuracy (Acc-n) of annotations by non-proficient speaker against annotations by proficient speakers.

Annotation quality For a subset of audited languages from CCAligned and OSCAR 2019 we measure the accuracy (Acc) of the labels assigned by non-proficient speakers against the labels assigned by proficient speakers for all audited sentences. This can be understood as a directed measure of annotator agreement for the special case where one rater is an expert and the other is not. Results for varying label granularity are reported in Tables 6.3 and 6.4. For $n = 6$ all classes of the taxonomy were distinguished, for $n = 4$ the `C` subclasses were combined, and for $n = 2$ it is binary decision between `C` and the rest of the error classes. With the full 6-class taxonomy (Acc-6) we find a mean accuracy of 0.66 for CCAligned audits, and 0.98 for OSCAR audits. With a binary taxonomy (Acc-2) distinguishing `C` from the rest, the accuracy further increases to 0.79 for CCAligned. This provides strong evidence that good quality annotations are not limited to those proficient in a language.

However, the significant drop of accuracy for finer-grained labels hints at that our taxonomy can be further improved, especially for parallel sentences. The error taxonomy lacks at least one category of error, namely "correct/in-language but unnatural". Similarly, the definition of "correct-short" and "correct-boilerplate" were not understood equally by all annotators and the concept of "correct-short" has potential issues for agglutinative languages like Turkish. Finally, it was unclear what to do with related dialects, e.g. when a sentence is "almost correct but wrong dialect" or when it is unclear which dialect a sentence belongs to. We recommend including these categories for future audits

6.1.3 Automatic Filtering

Given the frequency of `WL` and `NL` annotations, it might be tempting to use open-source LangID models to post-filter data on a per-sentence(-pair) level, as OSCAR does. Unfortunately, this turns out to have its own issues.

Sentence-level n-gram LangID filtering We classify all sentence pairs of CCAligned with CLD3, an n-gram based LangID model. By comparing its predictions to the audit labels, we evaluate its quality on the subset of annotated samples: the classifier should detect both correct languages when the pair is annotated as `C` and `X`, and should detect incorrect languages in the pair when `WL` and `NL`. On this task, the CLD3 classifier achieves an average precision of only 40.6%.

Sentence-level Transformer LangID filtering N-gram LangID models like CLD3 have known problems. However, Caswell et al. (2020) demonstrate that semi-supervised Transformer-based LangID models strongly out-perform them. We train a comparable Transformer-based LangID model and apply it to our annotated CCAligned data. We find that filtering noisy corpora ($< 50\%$ correct) on LangID for both source and target leads to gains in median precision, rising from 13.8% pre-filter to 43.9% post-filter. However, this comes at a steep cost of 77.5% loss in recall. The biggest winners were Lingala, whose precision climbs from 8% to 80%, and Oromo, which soars from 2% to 33% in-language. Both of these, however, come at the cost of losing 50% of the correct in-language sentences, being reduced from 22k sentences to 3k and 1k sentences respectively, which would likely be too small for building downstream models. The moral is that, at least at the current stage, there is no one-size-fits-all approach for sentence-level LangID filtering.

6.2 Dataset Mis-labeling

Standardized and unambiguous representations of language codes are important for practical data use and exchange. The standard used by most academic and industry

applications is BCP-47 (Phillips and Davis, 2005), which builds off the two-letter ISO639-2 codes and three-letter ISO639-3 codes, but also allows to add subtags for scripts (e.g. Hindi in Latin script: `hi-Latn`) or regional varieties (e.g. French spoken in Canada: `fr-CA`). It would enhance transparency and interoperability if adopted consistently, especially with growing language diversity in NLP.

We find a variety of errors and inconsistencies in language code usage, ranging from serious mislabelings to small transgressions against standard conventions. For this analysis, we also include the JW300 (Agić and Vulić, 2019) dataset, a multilingual dataset crawled from `jw.org`. In summary, we find 8 nonstandard codes in CCAligned, 3 in OSCAR 2019, 1 in mC4, 1 in WikiMatrix, and 70 in JW300, for 83 in total. This does not include the 59 codes affected by superset issues. Full details are given in Appendix C.1.

Inconsistent Language Codes One common issue is simply using nonstandard or invented codes. For example, CCAligned uses only two-letter codes, so when the BCP-47 code for a language is three letters it is either shortened (e.g. `zza` → `zz`) or invented (`shn` → `qa`). Similarly, OSCAR 2019 contains data labeled as `als` (BCP-47 for Tosk Albanian) that is actually in `gsw` (Alemannic).[5] 22 additional language codes in JW300 have similar issues, including 12 codes that start with `jw_` but are not Javanese.

False Sign Languages 12% (48/417) of JW300 carry language codes for sign languages. Instead of sign language transcripts they are texts in another high resource language, mostly English or Spanish—for example, the `en-zsl` (Zambian sign language) data is actually English-English parallel data (copies), details in Appendix C.1. This was likely caused by videos with sign language interpretation embedded on the crawled websites.[6]

Mysterious supersets When datasets contain language codes that are supersets of other language codes, it is difficult to determine which particular language the text contains. WikiMatrix has Serbian (`sr`), Croatian (`hr`), Bosnian (`bs`), and Serbo-Croatian (`sh`)—their superset.[7] The issue of codes that are supersets of others is common enough to include a small table dedicated to it (Appendix Table C.1). In some cases this may not be an issue, as with Arabic, where `ar` conventionally refers to Modern Standard Arabic, even though the code technically encompasses all dialects. In many cases, the nature of the data in the superset code remains a mystery.

[5] This is a result of the language code used by the Alemannic Wikipedia and affects any corpus or tool that uses Wikipedia data without correcting for this, like FastText.
[6] Kudos to Rebecca Knowles for this explanation.
[7] `https://iso639-3.sil.org/code/hbs`

Deprecated codes Finally, there are several deprecated codes that are used: `sh` in Wikimatrix, `iw` in mC4, `sh` and `eml` in OSCAR 2019, and `daf` in JW300.

6.3 Risks of Low-Quality Data

Low quality in downstream applications Text corpora today are building blocks for many downstream NLP applications like question answering and text summarization—for instance, a common approach is to first train translation models on such data and then automatically translate training data for downstream models (Conneau et al., 2018). If the data used for the original systems is flawed, derived technology may fail for those languages far down the line without knowing the causes. This risk of undesired downstream effects calls for future studies with a careful treatment of intertwined effects such as data size and domain, language-specific phenomena, evaluation data and metric biases. To give the reader a brief glimpse of the impact of data quality for the example of translation, we compare the `C`% metric from our audit with the translation quality (sentencepiece-BLEU, spBLEU) of the multilingual translation model M2M124 for 124 languages (Goyal et al., 2021). It was trained on WikiMatrix and CCAligned, and similar data collected with the same tools, which we expect to show similar biases. Translation quality is evaluated on the trusted, human-translated FloReS benchmark (Goyal et al., 2021). For the 21 languages present in both the audit and the FloReS benchmark, we found a positive correlation (Spearman) between the data quality scores and spBLEU of $\rho = 0.44$ ($p = 0.041$). This is not as large as the correlation with data size ($\rho = 0.66$, $p = 0.00078$), but it nonetheless helps to explain translation quality—the correlation between the product of `C`% and data size (in other words, the expected total number of good sentences in the dataset), is the highest yet, with a value of $\rho = 0.73$ ($p = 0.00013$).[8]

Representation washing Since there are datasets which contain many low-resource languages, the community may feel a sense of progress and growing equity, despite the actual quality of the resources for these languages. Similarly, if low-quality datasets are used as benchmarks they may exaggerate model performance, making low-resource NLP appear more solved than it is—or conversely, if models perform poorly when trained with such data, it may be wrongly assumed that the task of learning models for these languages is harder than it actually is or infeasible given current resources. These effects could result in productive effort being redirected away from these tasks and languages.

[8] For the translation from English, BLEU scores are less comparable but the trend holds nonetheless, with values of ($\rho = 0.32$, $p = 0.14$), ($\rho = 0.74$, $p = 0.000078$), and ($\rho = 0.80$, $p = 0.0000087$) respectively.

en	The prime minister of the **UK** is **Boris Johnson**.
nl	De minister-president van **Nederland** is **Mark Rutte**.
	en: The prime minister of the Netherlands is Mark Rutte.
en	**24 March** 2018
pt	**14 Novembro** 2018
	en: 14 November 2018
en	The current local time in **Sarasota** is **89** minutes.
nn	Den lokale tiden i **Miami** er **86** minutt.
	en: The local time in Miami is 86 minutes.
en	In **1932** the highway was extended **north to LA**.
bar	**1938** is de Autobahn bei **Inglstod** fertig gstellt.
	en: The highway near Inglstod was completed in 1938.

Table 6.5: Examples of "parallel" data where the translation has a different meaning than the source, but the form looks the same. (We added translations of the non-English side.) Such data may encourage hallucinations of fake "facts".

Trust in incorrect "facts" We found many instances of parallel-looking sentences that are structurally and semantically similar, but not factually correct translations (Table 6.5). They can cause models to produce plausible "translations" that are factually wrong, but users may still trust them (*algorithmic trust*) without verifying the information. Similarly, *automation bias* (Skitka et al., 1999), referring to humans favoring decisions made by automated systems over decisions made by humans, might amplify the issues of inaccurate translations caused by misalignments.

6.4 Future Work and Recommendations

Of the five multilingual corpora evaluated, we consistently found severe issues with quality, especially in the lower-resource languages. We rated samples of 205 languages, and found that 87 of them had under 50% usable data, with a full 15 languages at 0% in-language. We furthermore found consistent issues with mislabeled data and nonstandard language codes, particularly in the JW300 dataset, and identified 83 affected corpora, at least 48 of which were entirely spurious (Section 6.2). While there might have been anecdotal evidence of insufficient quality for some datasets, the majority of these quality issues had not been reported, nor been investigated in depth. These issues might go unnoticed for languages that are not represented in the evaluation of the crawling methods, and cause harm in downstream applications (Khayrallah and Koehn, 2018).

There are a variety of ways to improve both the ease and accuracy of human evaluation, as well a few classes of issues we ignored in this paper, like close dialects. Ideally we would like to build a standard suite of automatic metrics for datasets, but more research is necessary to determine what the appropriate metrics would

be. One important area missing from our analyses however is the estimated portion of a dataset which has been generated by MT (Rarrick et al., 2011), LM systems, or bots/templates, as for example in the analysis of C4 (Dodge et al., 2021). The information captured in machine-generated content might still be useful for modeling, but might falsely overrepresent typical generation patterns and introduce linguistic errors or unnatural artifacts.

We therefore strongly recommend looking at samples of any dataset before using it or releasing it to the public. As we have shown, one does not need to be proficient in a language to see when there are serious quality issues, and a quick scan of 100 sentences can be sufficient to detect major problems. Moreover, going through and annotating a small sample of data can bring actionable insights about new ways to filter or use it.

If data quality issues are found, a wide variety of techniques can be explored, like filtering on length-ratio, LangID, TF-IDF wordlists (Caswell et al., 2020) or dictionaries (Kamholz et al., 2014); to neural approaches like LM scoring (Axelrod et al., 2011; Moore and Lewis, 2010; Wang et al., 2018). Unfortunately, none of these provides a quick and easy fix, especially for low-resource languages—data cleaning is no trivial task!

Noisy datasets are by no means useless, at least if they contain some desirable content. Therefore, an alternative to filtering can be documentation (Bender et al., 2021). This can take the form of a per-language quality score and notes about known issues, a datasheet (Gebru et al., 2018) or nutrition label (Holland et al., 2018). However, we suggest researchers not release corpora with near-zero in-language content, as this may give the mistaken impression of usable resources.

Finally, we encourage the community to continue conducting evaluations and audits of public datasets—similar to system comparison papers.

6.5 Conclusions for the OSCAR Project

While the study described in chapter 5 showed encouraging results for the OSCAR 2019 corpus, a lot of concerns about the actual quality of the data remained unaddressed. This has addressed some of these concerns and actually showed promising results for the OSCAR corpus especially in comparison to the other four audited corpora, as OSCAR 2019 obtained the highest percentage of correct sentences as shown in table 6.2.

However, we also acknowledge that major issues remain to be addressed as has been pointed out here and more importantly, only 0.00004% of the corpus was actually audited here, meaning that potential issues with both the corpus and the pipeline might remain to be discovered. This collaboration marks thus a turning point for the OSCAR project, as it served as a platform and catalyst for both relaunching the project and start working on further versions of the corpus to the one originally

published in 2019 (Ortiz Suárez et al., 2019). The following two chapters will describe the creation of two subsequent versions of OSCAR that try to address some of the problems described here and some others that were pointed by the users of the project at both the corpus and the pipeline level.

7 Ungoliant: The Second OSCAR pipeline

> In which we present the work of Abadji et al. (2021), who after the evaluations discussed in the previous two chapters, completely rewrote the original OSCAR's `goclassy` pipeline, added features to the corpus such as metadata extraction and published the second version of the OSCAR corpus now known as OSCAR 21.09.[1]

As discussed in previous chapters, OSCAR 2019 was generated from the plain text data extracts (WET files) of the November 2018 Common Crawl dump, which was distributed in the form of 56,000 *shards*, that were then filtered and classified by language (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020b). OSCAR 2019 is now available for research through the Huma-Num servers[2] in Europe and for the public at large through Hugging Face's Datasets Hub[3] where it now has more than 15 thousands downloads.

OSCAR 2019 came in four different versions, each one intended for different tasks. These versions were either *unshuffled* or *shuffled* (that is, for each language, lines have been shuffled, destroying record and thus document integrity), and *non-deduplicated* or *deduplicated* (since duplicate lines account for more than half of the total data[4] generated by the pipeline). For the unshuffled versions, each language file contained paragraphs that came from the same record, and each paragraph is separated by a newline.

Simply put, OSCAR 2019 was composed of single language files that contained textual data (`ta.txt` for the Tamil language, for example). However, due to the often huge sizes of these files, and subsequently the impracticality of storage and distribution, OSCAR 2019 files were split and compressed in equally sized parts.

[1] Contributions: I designed most of the experiments and comparisons to be made between the two pipelines, I downloaded and prepared Common Crawl for extraction and I actively participated in the writing of the scientific article.

[2] https://oscar-corpus.com/post/oscar-2019/

[3] https://huggingface.co/datasets/oscar

[4] OSCAR-orig: 6.3TB, OSCAR-dedup: 3.2TB

However, but OSCAR 2019 and its pipeline came with a number of limitations, which we will discuss in the following sections, and we will try to start fixing in this and the following chapter.

7.1 Limitations of the OSCAR 2019 Corpus and its Generation Pipeline

7.1.1 OSCAR 2019

OSCAR 2019 was inherently linked to its generation pipeline, and as such its quality partly depended on the pipeline's quality. While OSCAR 2019 was considered to be one of the cleanest multilingual corpora available as discussed in the previous chapter and in (Caswell et al., 2020), several problems had been described, and the state of the publicly available code raised questions about maintenance and maintenability of the pipeline itself. Apart from the fact that its content dated back to 2018, some of the problems with OSCAR 2019 include:

- **Language label mismatches and inconsistencies**, which occurs earlier in the pipeline and would be fixable downstream,
- **Representation washing** as defined by Kreutzer et al. (2022), whereby low resource languages, while present in the corpus, are of a significantly lower quality than higher resource languages without any quality metric available publicly.

Moreover, the more recent dumps of Common Crawl in 2021 contain more than 64,000 shards (almost 10,000 more than the dump used for OSCAR 2019). Furthermore, each of these shards is composed of numerous records, and each record holds textual content along with metadata. While Common Crawl shards hold document-level metadata that could be useful downstream, they were com discarded and do not appear in OSCAR 2019, whereas other corpora generated from Common Crawl do include them, e.g. CCNet (Wenzek et al., 2020). This limits OSCAR 2019 users to the textual content only, whereas metadata could have been distributed along with the corpus itself.

7.1.2 goclassy

OSCAR 2019 was built using `goclassy`, a high-performance asynchronous pipeline written in Go described in Chapter 4. However, it suffered from several caveats that makes the re-generation and update of the corpus relatively complex in practice.

While `goclassy`'s source code was easily readable thanks to the choice of an uncluttered programming language and a pragmatic approach, the lack of structure

in both the source and the project itself made `goclassy` difficult to extend and maintain.

The pipeline was not functional out-of-the-box, as the user had to provide the compressed shards from CommonCrawl, manually install FastText (Joulin et al., 2016; Joulin et al., 2017) and create specific directories by themselves, since only partial instructions are given in the supplied README file.

As described in Chapter 4, `goclassy` also made heavy use of I/O, as data was saved and loaded repeatedly between steps; as an example, the identification step stored language identification data and individual sentences in two files, before generating the final files (one per language). Despite these limitations, `goclassy`'s performance remained acceptable mainly due to Go's emphasis on easy and efficient parallelization and inherent speed. The pipeline for instance used clever handling of file descriptors and employed extensive buffering, which limited I/O calls cost in some parts.

7.2 Building a New Version of the OSCAR Corpus

Having identified some shortcomings of both OSCAR 2019 and its pipeline, `goclassy`, we decided to restart the OSCAR project by completely rewriting our pipeline. To that end, we introduce *Ungoliant*, a new corpus generation pipeline that, like `goclassy`, creates a large-scale multilingual text corpus from a Common Crawl dump. However, contrarily to `goclassy`, Ungoliant is fully modular, better structured, and highly parametrizable; thereby allowing comparisons between several parallelization strategies. A specific effort was put in testing and documentation. Parts of Ungoliant are heavily inspired by `goclassy`, although for its implementation we decided to use Rust rather than Go, which is often considered to be a faster more low level programming language.[5]

We also use Ungoliant to generate a new version of the OSCAR corpus from a more recent Common Crawl dump. The new corpus includes metadata information while retaining backward compatibility with the OSCAR 2019 corpus.

7.2.1 Ungoliant

Rationale and Scope

While Ungoliant is heavily inspired by `goclassy`, it provides a better set of tools to download, process, filter and aggregate textual and contextual data from Common Crawl. These operations can be sequential, parallel or both, depending on contexts and performance requirements.

We provide both batch and streaming processing, so that the whole pipeline could be run either online, with every step running on streams of data, or offline, with

[5] https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/rust-go.html

Platform	#shards	goclassy	Ungoliant	Approx. speedup
Desktop	1	30s	13s	×2.3
	10	3m6s	2m12s	×1.3
	25	9m10s	5m47s	×1.5
HPC	1	40s	6s	×6.6
	25	2m40s	1m6s	×2.4
	100	7m59s	4m14s	×1.8

Table 7.1: Comparison of approximate generation times depending on platform and number of shards.

every step running on tangible files, or a mix of both, using already downloaded Common Crawl dumps but streaming the rest of the process. Moreover, we embed numerous filtering and deduplication utilities directly inside Ungoliant, making these features available for pipeline composition and post-processing.

Ungoliant features a loosely defined pipeline interface, on which we re-implement `goclassy`'s one, while improving performance by threading more aggressively and avoiding I/O where it is not necessary: While `goclassy` uses intermediate files for tags and sentences, we try to keep everything in memory in order to avoid losing time loading or writing files. The Rust language provides constructs that helps us build complex abstractions and pipelines while limiting proactive file I/O or computing, since nearly all the reimplemented pipeline is built around lazy evaluation. File I/O is only used when loading shards, and when writing sentences in language files.

Through benchmarking we found that the best parallelization strategy is to use rayon,[6] a work-stealing (Blumofe and Leiserson, 1999) parallel and concurrent library enabling massive parallelization. We parallelize on shard-, record- and sentence-level processing.

To evaluate Ungoliant performance, we run both `goclassy` and Ungoliant's implementation on 1, 10, 25 and 100 Common Crawl shards both on a middle-range laptop computer (i5-7200u, 8 GB RAM, NVMe SSD) and a HPC node (Xeon 5218 (64 Threads), 180 GB RAM). Results are shown in Table 7.1.

Ungoliant performs better than `goclassy` on all tasks, independently of the platform or number of shards processed. However, we can note that Ungoliant's speedup is higher on short tasks, which is explained by its aggressive multithreading strategy, while `goclassy` uses a record-scope multithreading at its finest granularity.

7.2.2 Iterating on the goclassy Pipeline

Common Crawl dumps contain metadata that hold useful information such as related records, recognized language(s), or origin URLs. Since OSCAR's 2019 pipeline discarded metadata and sentences could be shuffled, we lost the ability to investigate

[6] https://github.com/rayon-rs/rayon

the metadata itself, as well as working on potentially multilingual documents, since we separated text from metadata.

The new pipeline (and the resulting new corpus schema) aims to establish a first link between textual data and metadata from Common Crawl, while staying backward compatible with the existing OSCAR 2019 schema.

In other words, switching from the original OSCAR 2019 corpus and the newly generated one should be a drop-in operation.

Metadata Extraction and Linking

Our choice of keeping the corpus backward compatible with the original OSCAR 2019 introduces changes in the way the corpus is generated, namely regarding metadata: a record's body is composed of sentences that *aren't guaranteed to be of the same language*. Since OSCAR merges sentences from multiple records into a single file, special attention has to be paid to the metadata dispatch too.

Approaches to tackle this problem range from (1) storing all metadata in a single location to (2) having language-specific metadata files that contain the metadata for each line in the language file.

Both (1) and (2) have their strengths and weaknesses, namely:

1. Having all metadata at the same place may facilitate wide queries about whole metadata, but at a cost of a very large size (which harms both accessibility and performance).

2. Getting the metadata for a given line is fast since line numbers are synchronized, but there is repeated information and a potentially important increase in size.

We thus choose a hybrid approach which keeps metadata local to each language, while trying to limit the information repetition by keeping an entry by group of *chunks* rather than by line, where a *chunk* is a series of contiguous sentences that share the same language from the same document.

An overview of the pipeline can be seen in Figure 7.1, where we depict Ungoliant at a macro level in the first part of the figure, and where we also give a more precise view on record processing and metadata extraction in the second half of the figure.

Metadata is distributed via JSON-encoded files holding an ordered list of metadata entries, along with offsets (o) and paragraph lengths (l), enabling any user to get the content of a said metadata by querying for lines $(o, o + l]$ in the content file.

This approach still has drawbacks, in particular when looking for the corresponding metadata of a given sentence/paragraph, where one has to perform a search on the metadata file, or when working with multilingual documents. Another important drawback is the resulting cost of potentially merging back numerous language parts: Since metadata query is offset-based, merging back metadata files implies updating those offsets.

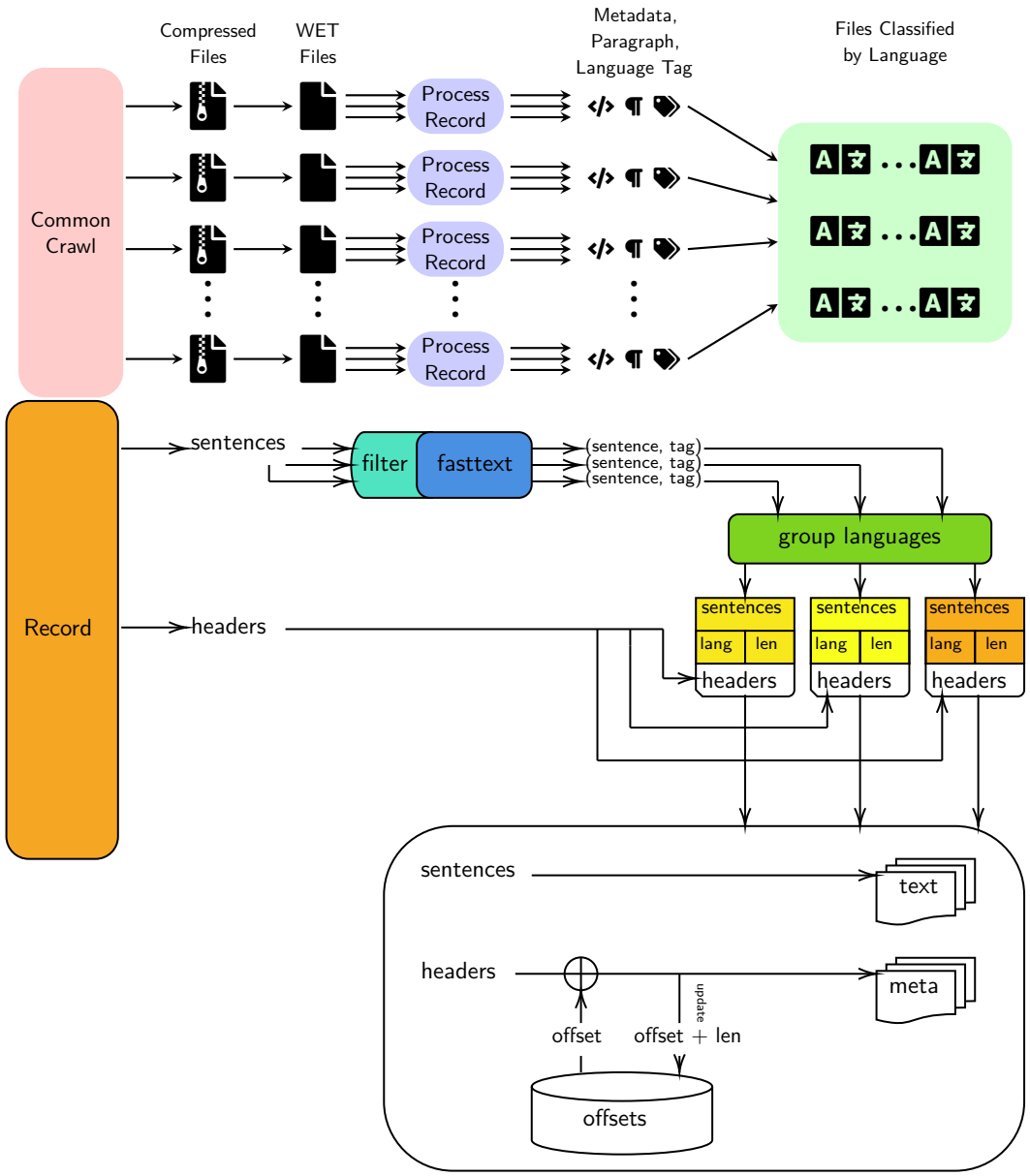


Figure 7.1: Record processing with metadata extraction. Headers are kept aside while sentences are identified and grouped into same-language bins. Headers are then cloned for each bin, and are sequentially stamped with an offset that is recorded for the whole operation, and written to disk into text and metadata files by language.

Platform	#shards	Without Metadata	With Metadata	Speedup
Desktop	1	13s	12s	×1.1
	10	2m12s	1m55s	×1.1
	25	5m47s	4m50s	×1.2
HPC	1	6s	7s	×0.9
	25	1m6s	1m12s	×0.9
	100	4m14s	4m36s	×0.9

Table 7.2: Comparison of approximate generation times with and without metadata generation.

Having paragraphs and metadata linked by offsets in a highly parallelized pipeline implies to take special care at the offset level. The solution is to use shard-scoped offsets (starting from 0 for each language), and to keep global offsets protected by a mutex guard. This way, when a given shard is done processing and is ready to be written on disk, we convert shard-scoped offsets to global-scoped ones, update the global-scoped ones and then write text and metadata on disk.

We compare running times for the reimplementation of the `goclassy` pipeline, and our new pipeline adding metadata extraction, using both desktop and HPC contexts. The results are reported in Table 7.2.

Metadata generation does not seem to influence generation time dramatically. However, we can notice a slight performance difference between HPC and Desktop contexts. These differences may lie in the storage medium differences, I/O layout, or algorithmic peculiarities benefiting desktop contexts because of other bottlenecks.

7.2.3 Characteristics of the OSCAR 21.09 Corpus

We evaluate the newly generated OSCAR 21.09 corpus (published in September 2021[7]), assessing its ability to reflect events that occurred after the publication of OSCAR 2019, that is, events that occurred after November 2018, and we detail the metadata format and potential use.

Comparison with OSCAR 2019

While it is expected that our new corpus has a larger file size than OSCAR 2019 since Common Crawl itself grew from 7.42 TB to 8.06 TB, metadata quickly adds up and accounts for nearly 15% of the total uncompressed data in OSCAR 21.09.

The size difference is not the same for each language, and while the corpus as a whole is bigger now, some languages are smaller than they were before.

Results show that already largely represented languages gain more and more data (like the English language, which constituted more than a third of the original

[7] https://oscar-corpus.com/post/oscar-v21-09/

OSCAR Version	Common Crawl	OSCAR (dedup)	Metadata	Total (increase)
2019	7.42TB	6.3TB (3.2TB)	N/A	6.3TB
21.09	8.06TB	7.2TB (3.3TB)	1.2TB	8.4TB (+33%)

Table 7.3: Comparison of the size of the Common Crawl dumps and their corresponding OSCAR sizes between the 2019 and the 21.09 versions. Compressed (Common Crawl) sources are from November 2018 and February 2021 dumps. Total is Textual + Metadata without deduplication.

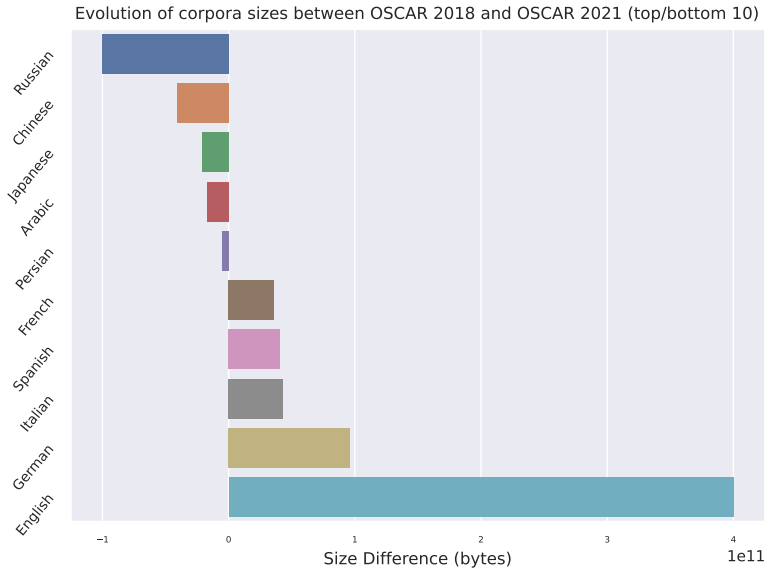


Figure 7.2: Comparison of language size (in bytes) between OSCAR 2018 and OSCAR 2021 (top/bottom 5 only).

OSCAR 2019), except for the Russian language which loses approximately 100Gb of textual content. These results are summarized in Figure 7.2.

However, in a context where the number of languages is very high (higher than 150) and of varying sizes, evolution can't be analyzed via a mere size evaluation. By computing, for each language, the relative size difference between the 2019 and 21.09 releases of OSCAR, less resourced languages do appear, hinting at a better representation of some of them. These results can be found in Figure 7.3.

Note nonetheless that numerous languages have been omitted from Figure 7.3, either:

- because they were present in the original OSCAR 2019 and are now absent (*Central Bikol* and *Cantonese*)
- or because they were absent in the original OSCAR 2019 and are now present (*Manx*, *Rusyn*, *Scots* and *West Flemish*)

Precautions have to be taken when using these corpora and further work has to be done to correctly assess the quality of low-to-mid resource languages in order to better reflect the quality of each corpus to the OSCAR users. Some sub-corpora exhibited either a particularly low number of sentences or just very low quality data, and as such they are not really usable in practice. However, they still account for a language in the total language count of both the original OSCAR 2019 and the new OSCAR 21.09.

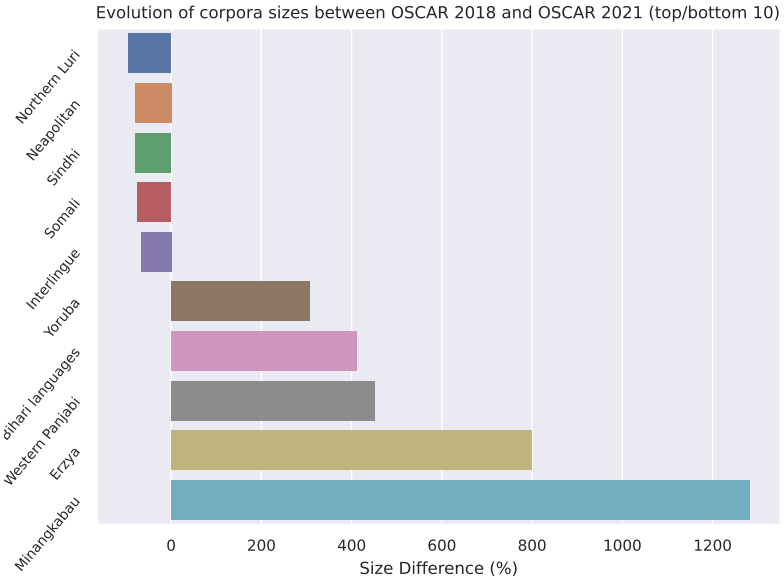


Figure 7.3: Comparison of language percentage between OSCAR 2018 and OSCAR 2021 (top/bottom 5 only).

Metadata

Metadata provides new contextual data that is useful to evaluate the corpus and draw metrics.

The total size of metadata is 1.2 TB, ranging from 4Kb to 500Gb, depending on the number of lines. Relative size varies from 100% to 20%, diminishing with the textual data size, which is expected.

Our choice of keeping metadata aside from the main content adds some complexity when working with both textual and contextual data:

- When trying to get the metadata of given sentence, one has to get the line number k, then sequentially (or use a search algorithm since offsets are sorted) look for the record (with offset o and length l), where $k \in [o, o + l]$.

Language	Term	2018	2021
Arabic	Beirut port explosion	0	31
Burmese*	Min Aung Hlaing	387	3439
English	Obama	30039	27639
English	Biden	990	19299
French	Yellow Vests	2	96

Table 7.4: Comparison of occurrences of news-related terms between OSCAR and our corpus in a sample of 100 Common Crawl shards. For the Burmese language, we use the whole 2018 and 2021 corpus since it is a low resource language. Terms are translated to the target language.

- Looking for lines corresponding to a particular metadata entry is easier: one has to read the textual file, skipping until the o-th line, then read l lines.

Presence of events

Using a sample of five sub-corpora, we perform a simple search of terms in order to assess and compare the presence of pre- and post- 2018 events and persons in both corpora. Terms and frequency are grouped in Table 7.4.

Our corpus keeps around the same number of occurrences for pre-2018 events or public figures such as Barack Obama, while increasing the occurrence of people linked to more recent events (Joe Biden).

We include search terms linked to post-2018 events in French and Arabic which are smaller corpora (resp. 200 and 80 GB), and in Burmese, a mid-resource language (approximately 2 GB). We observe a term occurrences evolution that reflects the linked events' timing and importance.

7.2.4 License

This new OSCAR 21.09 corpus is released under a research-only license that is compliant with the EU's exceptions for research in text and data mining. Contrarily to the original OSCAR 2019, no shuffled version of the corpus is distributed, instead we put in place an authentication system that allows us to verify that requests for the corpus come from research institutions. A contact form is also provided for independent researchers so that we can study their particular cases and determine if the utilization of the corpus corresponds to a legitimate research use.

Moreover, the introduction of metadata makes our corpus far more queryable, thus simplifying and speeding up the handling of take-down GDPR requests. For this reason, we release the complete set of metadata under a CC0 public domain license, so that any individual can check if their personal or even copyrighted data is in our new OSCAR 21.09 corpus and make a request accordingly.

7.3 Conclusion

Although the work presented in this particular chapter does not directly address some of the previous concerns raised by Caswell et al. (2020) and discussed in Chapter 6. We do believe that a more efficient, more modular and better documented pipeline is the first step in making the OSCAR project more approachable by other members of the NLP and Digital Humanities communities.

Moreover, we also believe that the addition of metadata to OSCAR is a big step towards improving the quality of its content as it will provide us and other researchers willing to use OSCAR with enough information to better explore, audit, annotate and filter the corpus.

In the next and final chapter of the OSCAR part in this thesis we will explore the question of document integrity which might be useful for researchers interested in document level tasks and which until now is not respected for Common Crawl records containing multilingual data. We will also continue improving Ungoliant and start using the metadata that we extract from the Common Crawl records to produce the first ever OSCAR annotations.

8 Towards a Cleaner Document-Oriented Annotated OSCAR Corpus

In which we present the work of Abadji et al. (2022), who continued improving over the second OSCAR pipeline *Ungoliant* by adding mechanisms to ensure document integrity, specially for multilingual records of Common Crawl, and also by adding the first methods for simple annotations of the OSCAR corpus that would allow users to more easily filter the data and obtain a cleaner dataset specially for language modeling applications.[1]

In this final chapter about the OSCAR project we present the first methods for adding simple annotators to the Ungoliant pipeline that build upon the improvements presented in the previous chapter and that actually allow us to finally start addressing some problems exposed in chapter 6 and in far more detail in (Caswell et al., 2020; Kreutzer et al., 2022). Moreover, we also introduce a new method for document level language classification that:

1. Is based on line-level language classification allowing us to hopefully preserve the classification quality that we saw in chapter 6.

2. Allow us to respect document integrity such that we can establish a one to one correspondence between OSCAR documents and Common Crawl records.

3. Allows us to get multilingual documents that might one day serve as the basis of a parallel OSCAR corpus.

[1] Contributions: I devised the length based filter and the length based annotations, as well as the adult annotations based on a blocklist. I also came up with the method for document label language identification as well as the formalism presented in section 8.2. I proposed some of the comparisons we did between OSCAR versions. Finally, I actively participated in the writing of the scientific article.

8.1 Filtering

Previous OSCAR pipelines were line-oriented (where a line is defined as a string separated by `\n`, and often correspond to a paragraph), which meant that the highest filtering granularity were lines. Having a document-oriented corpus implies that:

- We must try to keep the document integrity, by altering it in a way that does not completely destroy its coherence.
- Operations on the document (filtering, identification, annotation) must take into account the document as a whole.

We aim to produce a corpus that is similar in size and quality to OSCAR 21.09, looking for a set of filters that limits the inclusion of short, noisy lines in documents, while keeping a sufficient quantity of data, especially for low- and mid-resource languages. Those filters either keep/discard a given document, or remove lines from the document body then keep it.

8.1.1 Header and footer filter

Similar to previous OSCAR pipelines, we use a length-based filter discarding short-lines. However, we restrict the removal on contiguous sequences of short lines that are located either at the head or at the tail of the document. In the following document, only the lines preceded by an exclamation point would be kept.

```
Home
Login
Sign Up
Welcome to my Website
! Lorem Ipsum Dolor Sit Amet ....
! Lorem Ipsum Dolor Sit Amet ....
! Lorem Ipsum Dolor Sit Amet ....
! Lorem Ipsum Dolor Sit Amet ....
Copyright Myself
Legal
Contact
```

The solution still has numerous drawbacks, especially when dealing with documents crawled from the internet, a source known to be extremely noisy and full of edge cases: Adding a long line at the very head and tail of the previous document would completely negate the benefits of the filter.

8.1.2 Short lines proportion filter

In order to refine the filtering process, we use a count-based filter that separates the data in two bins: One for short lines and one for long lines. The filter then checks which bin is bigger, and filters out documents where the short lines bin is bigger.

This filter may limit the impact of documents containing low-quality long lines at the head/tail, then a high number of short lines.

8.2 Identification

The backbone of the language identification process is similar to the one used in `goclassy` (Chapter 4) for the generation of OSCAR 2019 and Ungoliant (Chapter 7) for the generation of OSCAR 21.09. However, shifting to a document oriented corpus (with a single top-level identification per document) requires to infer the document identification, based on line identifications.

We define a document $\mathcal{D}$ as a pair $\mathcal{D} = (\mathcal{L}, \mathcal{G})$ where $\mathcal{L} = \{l_1, \dots, l_n\}$ is the set of lines (strings separated by `\n`) that constitute the document and $\mathcal{G} = \{g_1, \dots, g_m\}$[2] is the set of languages identified by FastText for the document $\mathcal{D}$. When FastText is not able to identify a language for a specific line, for instance because the confidence isn't higher than 0.8, we tag said line with the *No Identification Language* that we simply note by g_0. Furthermore, we define each line l_i in a document $\mathcal{D}$ as a triplet $l_k = (g_i, p_i, s_i)$ where g_i is the language identified by FastText with the highest confidence for the line l_i, p_i is said confidence and s_i is the size in bytes of the line l_i. We also note $|l_i| = s_i$, and we thus define the size $|\mathcal{D}|$ of a document $\mathcal{D}$ as

$$|\mathcal{D}| = \sum_{i=0}^{n} |l_i| = \sum_{i=0}^{n} s_i.$$

Moreover, for each identified language $g \in \mathcal{G}$ in a document containing n lines, we define its size $|g|$ as

$$|g| = \sum_{\{i|g_i=g\}} s_i.$$

Finally, for each language $g \in \mathcal{G}$ we can also compute its *overall weighted confidence* P throughout the document $\mathcal{D}$ as the following weighted mean:

$$P = |\mathcal{D}|^{-1} \sum_{\{i|g_i=g\}} s_i p_i.$$

8.2.1 Multilingual document identification

A document can contain lines in multiple languages for several reasons:

[2] Note that since FastText identifies one language by line, we always have $m \leq n$ for every document $\mathcal{D}$.

1. Identification mismatch, that can show up frequently, especially with languages that have significant vocabulary overlap (Czech and Slovak),
2. Crawl from a website where the interface is written in a language, and the body is written in another one,
3. Crawl from a translation page, where the same content is present in two (or more) different languages.

In these examples, we should aim to limit the presence of 1. and 2., while maximizing the presence of 3.: documents having a balanced set of lines per language. Thus, we decide to take a cautious approach, restricting the multilingual document identification test to the documents that:

- Have at least 5 lines,
- Have at most 5 different languages.

Next, we compute the *proportion* for each language $g_j \in \mathcal{G}$ in the document $\mathcal{D}$ defined as follows

$$\Pr_{g_j} = \frac{|g_j|}{|\mathcal{D}|},$$

including for the no identification language g_0.

A document $\mathcal{D}$ containing n lines is identified as multilingual if and only if:

$$\begin{cases} |g_j| \geq \dfrac{|\mathcal{D}|}{n+1} & \forall g_j \neq g_0, \text{and} \\ |g_0| \leq \dfrac{|\mathcal{D}|}{n+1} \end{cases}$$

As an example, a document holding $m = 3$ languages is multilingual if each language makes up at least $\frac{1}{m+1} = \frac{1}{4}$ of the document, and that there is at most $\frac{1}{4}$ of the document that is of unknown identification.

8.2.2 Monolingual identification

We begin by identifying each line, keeping in memory the language identified, the confidence of the identification, and the size of the line. We keep track of lines that have not been identified with a special token, and a confidence of 1.

If the document does not pass the multilingual check, we then take the largest represented language and compute its overall confidence P_j and use a minimum confidence threshold of 0.6 that is way lower than the previous pipelines (0.8). This is motivated by the following reason: The document-based filtering removes documents containing lines that could have been kept by former pipelines, thus reducing the size of the generated data.

Using a lower threshold could help getting lower-quality documents that still hold high-confidence lines in themselves.

8.3 Annotation

While the filtering and identification steps are lenient by using lower thresholds than the previous pipelines, we introduce annotations, as non-destructive filters that enable more precise downstream filtering for the corpus users, as well as a useful resource to quickly assess the quality of a corpus. Annotations enable more aggressive filters to be run, since the non-destructive nature of annotations can in turn be used to refine annotation filters.

Numerous annotations are available, and each document can have several ones at the same time.

8.3.1 Length-based annotations

Some simple annotations are added when documents don't meet certain length requirements:

- The document has a low (≤ 5) number of lines (*tiny*)
- The document has a high number ($\geq 50\%$) of short lines (*short_sentences*)

These annotations help to spot potentially tiny documents, where the line structure or the document size could negatively influence training tasks.

A third annotation checks the occurrence of short lines at the start of the document, and adds a *header* annotation if it is the case, indicating that low-quality content could be present at the start of the document.

A fourth annotation named *footer* works in the same way on the tail of the document.

8.3.2 Noise detection

Some documents make their way into the corpus while being extremely noisy or non-linguistic. As an example, source code can be found in English corpora because of the presence of English words in the source itself.

We use a filter that computes a ratio between letters and non-letters.

This filter is based on Unicode categories. We use categories *Lu*, *Ll*, *Lt*, *Lm*, *Lo*[3] for letters, and we add categories *Mn*, *Mc*, *Me*[4] for accents and diacritics.

A *noisy* annotation is added if the ratio passes a certain threshold, set to 0.5.

[3]Lu: Uppercase letter, Ll: Lowercase letter, Lt: Titlecase, Lm: Modifier, Lo: Other.
[4]Mn: Nonspacing mark, Ms: Spacing mark, Me: Enclosing mark.

8.3.3 Adult documents

We use the UT1 blocklist[5] as a base for adult content filtering. The UT1 blocklist is a collection of thematic blocklists (adult, gambling, blog, among others), usually utilized in internet access control for schools. The list is constituted and extended by both human and robots contributions (known indexes, search engines, exploration of already known addresses). The blocklist is updated twice to thrice a week by Fabrice Prigent.

Each folder contains URL and domain blocklists, enabling filtering of both websites that are centered around adult content, and websites hosting user-generated content that can be of adult nature (several social networks...).

The adult blocklist comprises roughly 3.7M records.

8.4 Corpus

We apply the aforementioned pipeline to the November/December 2021 crawl dump of Common Crawl. The result is a new corpus, OSCAR 22.01. While its structure is different from the previous OSCAR corpora (due to the choice of generating a document oriented corpus), we have attempted to compare the two corpora, especially in terms of size and news-related topic presence and recall. We also evaluate the occurrence and pertinence of the annotations.

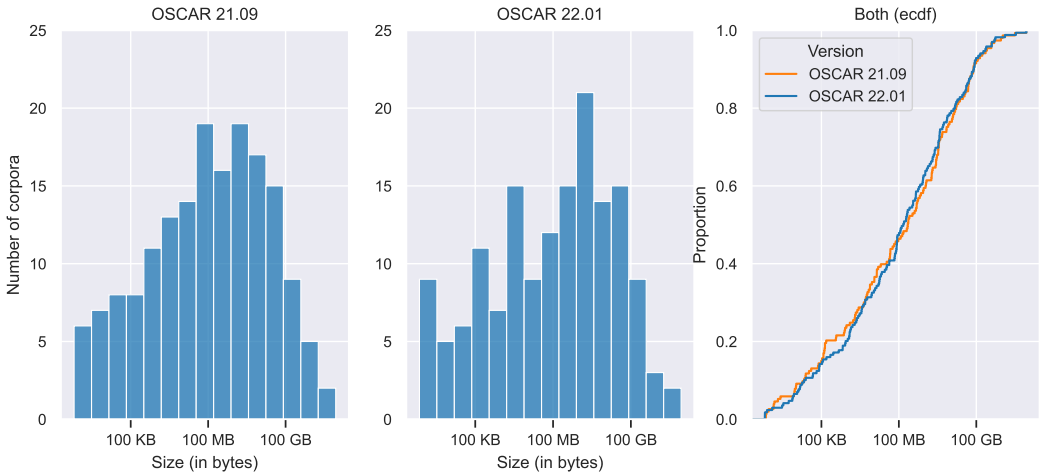


Figure 8.1: Corpus size distribution between OSCAR 21.09 and 22.01

[5] https://dsi.ut-capitole.fr/blacklists/.

8.4.1 Comparison with OSCAR 21.09

Size distribution

The data layout of OSCAR 22.01 may limit the relevance of raw size comparisons, since metadata are larger (annotations and line identifications were not present in previous OSCAR Corpora), and fused with textual data (metadata were distributed in separate files for OSCAR 21.09).

However, comparing the distribution of corpus sizes may help us ensure that the new corpus has a size distribution similar to the older one.

We compare the distribution of the sub-corpora sizes between OSCAR 21.09 and OSCAR 22.01 in figure 8.1. We see that while the overall distribution is similar, the lower end of the distribution has more variance: The $[0\text{B}, 100\text{KB})$ range shows more corpora at its bounds than at its center. Furthermore, we also plot the empirical cumulative density function, that helps to assert the distribution similarity between OSCAR 21.09 and OSCAR 22.01.

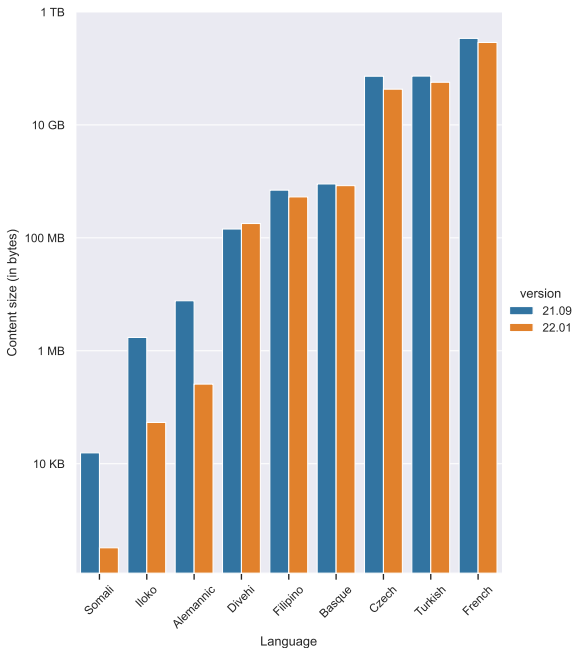


Figure 8.2: Content size comparison of selected languages in OSCAR 22.01 versus OSCAR 21.09

We also select three low-resourced languages, three mid-resourced languages and three high-resources languages and compare their content (that is, textual data excluding metadata) between OSCAR 22.01 and OSCAR 21.09. Comparison is shown

in figure 8.2. While the overall sizes of these corpora have slightly decreased, the sizes of the mid and high resource languages are similar enough.

Size differences in low-resource languages

The low-sized corpora exhibit important size changes. As an example, the Alemannic German corpus went from 7MB to 360KB between OSCAR 21.09 and OSCAR 22.01. This size decrease can be explained by the way the document identification works: by reasoning at a document level, documents containing a majority of German identified lines and a minority of Alemannic German identified lines will be identified as a German document, whereas previous OSCAR pipelines would have separated the lines and increase the size of the Alemannic German corpus.

By extracting the lines identified as Alemannic from the German corpus, we get around 30MB of data, which could constitute an Alemannic corpus with a size comparable to the OSCAR 21.09 Alemannic corpus after confidence and length based filtering.

This situation can, in a way, help us investigate the cases of linguistic proximity, where languages have a lexical overlap: When a line identified as Alemannic German is found inside a document that has been identified as German:

1. Is the line in German, and it is an identification error?
2. Is the line in Alemannic German, in a document that is in German? (ex: A German website related to the Alemannic German language)
3. Is the whole document in Alemannic German, and the identification classified the majority of Alemannic as German?

Those three cases can arise and may help to enhance the detection of a said language, by finding (1) identification mismatches, hoping that these cases would improve identification after training, or (3), after verification by a speaker of the language, state that the whole document is in Alemannic. The new data collected could in turn be used to improve language detection.

New themes

As OSCAR 22.01 is based on the November/December 2021 dump (compared to OSCAR 21.09, based on the February 2021 dump), the corpus should include data related to events contemporary to February 2021. We conduct a simple word search similar to the one conducted for the generation of OSCAR 21.09 (Abadji et al., 2021), using both old and new events, in order to give a rough idea of both the actuality and the memory of the corpus.

We see that the events and terms related to events predating February 2021 are still present in the corpus, but have a lower count that nevertheless remains in the same

Language	Term	21.09	22.01
Arabic	Beirut port explosion	31	13
Burmese*	Min Aung Hlaing	3439	2736
English	Obama	27639	8697
English	Biden	19299	8232
English	Omicron	131	417
French	Yellow Vests	96	73
Spanish	Aborto	1504	572

Table 8.1: Comparison of occurrences of news-related terms between OSCAR and our corpus in a sample of 100 Common Crawl shards.
*: For the Burmese language, we use the whole 21.09 and 22.01 corpus since it has less than 100 sentences in both subcorpora. Terms are translated in the corpus language.

order of magnitude. We also count the occurrences of the term Omicron, related to the Omicron variant, and observe that the term has a higher count on the 21.01 sample.

Absence of deduplication

Contrary to OSCAR 21.09, we do not distribute a deduplicated version of the majority of OSCAR 22.01.

The line-level deduplication of documents would have destroyed the integrity of documents themselves, hampering human readability and even sequential sentence sense. We can imagine having forum discussions' sense destroyed because of identical responses, or song lyrics being altered.

Moreover, the similarity-based document-level deduplication procedure is very costly in terms of computing power and time (Gao et al., 2020).

We make the choice of distributing a non deduplicated version of OSCAR along with a deduplicated, line oriented version of the English corpus, while encouraging the use of deduplication in the context of training language models (Lee et al., 2021). A line-level deduplication tool will be available as part of the OSCAR toolkit[6]. We will also distribute a deduplicated version of the English part of OSCAR 22.01, with a data layout similar to OSCAR 21.09 corpora.

[6] https://github.com/oscar-corpus/oscar-tools

8.4.2 Annotations

Raw stats

Annotations help us to infer the composition of the corpora: The *tiny*, *short_sentences* and especially *noisy* annotations may indicate documents of a varying poor quality, with *noisy* being the worst.

Also, comparing corpora annotation distributions, especially related to their size, could highlight potentially very low quality corpora. This semi-automated quality checking process could be used to label corpora where data quality is bad.

We select 3 low-resource (≃100KB), 3 mid-resource (≃100MB) and 3 high-resource (≃100GB) languages and plot the number of documents per annotation, adding a *total* legend for the total document count and a *clean* legend for documents that do not have any annotation. We then plot the counts for each resource group using adapted scales.

We observe that the annotation distribution is similar for each resource group, but that the lower resourced languages have a higher proportion of documents annotated with *short_sentences* and *tiny*.

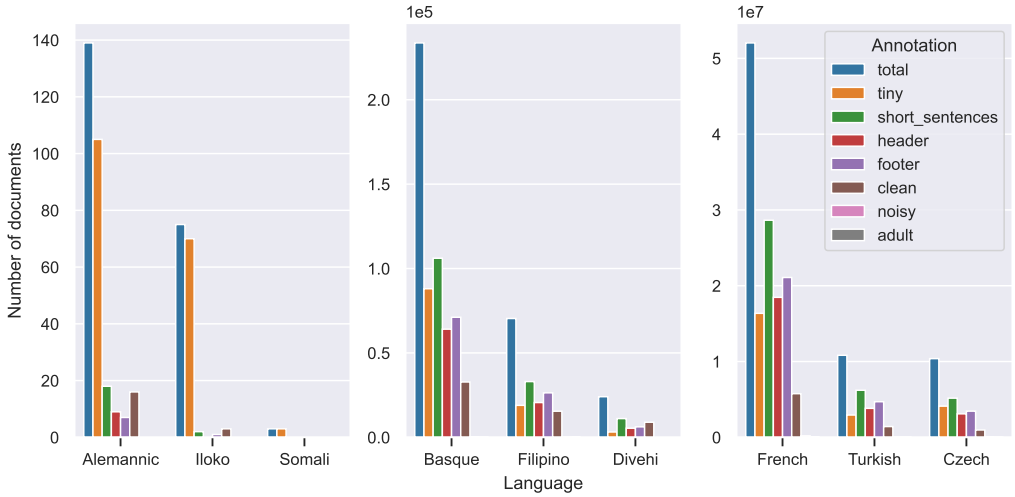


Figure 8.3: Annotation count in selected low, mid and high resource languages (scales are adapted to corpus size)

In order to better compare the resource groups, we display the annotation distribution in a heat map (figure 8.4). We notice important differences between low and mid/high resource groups. A very large proportion of the low resource group is annotated as *tiny* while simultaneously detaining few documents annotated *short_sentences*, indicating the presence of long sentences within documents with a low number of sentences.



Figure 8.4: Heat map of annotation distributions in selected low, mid and high resource languages.

Multilinguality

The OSCAR 22.01 Corpus also contains a multilingual corpus, composed of documents holding lines in multiple languages. Each document contains at least 2 languages, and at most 5.

We check the co-occurrence of languages, highlighting the coupling of language tuples. These tuples may highlight either linguistic similarity (Czech and Slovak, Russian and Belarusian) and subsequent poor classification, errors or languages commonly found together on documents. Due to the number of languages and the sparsity of the data, we show the language couples with a number of documents greater than 20 000 (Figure 8.5).

We also note the presence of English in a high number of documents. This could be explained by boilerplate content in web pages, such as menu headers or footers.

Using the clean annotation filter on the multilingual corpus may help to retrieve the highest quality multilingual documents.

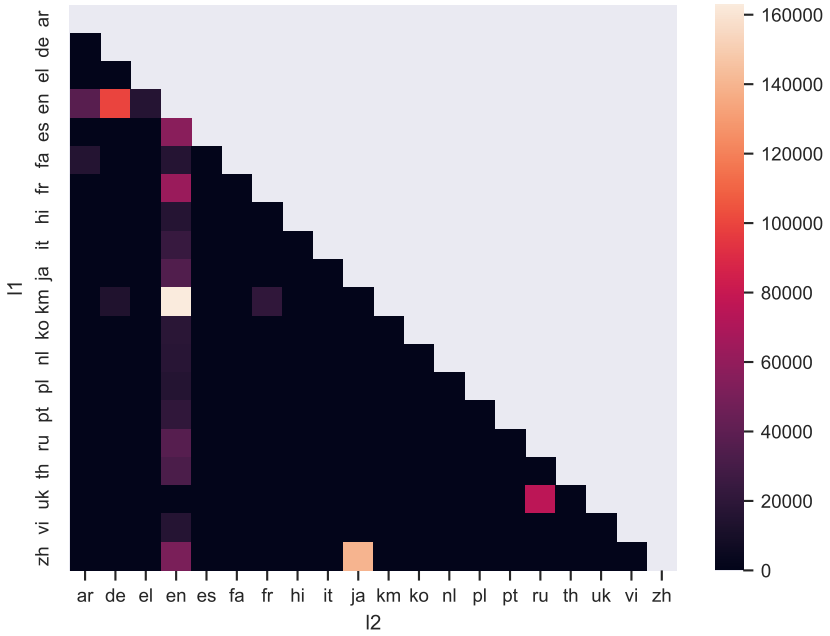


Figure 8.5: Count of $(l1, l2)$ language tuples in the multilingual corpus. Languages tuples with less than 20,000 occurrences are not shown.

Clean documents

We also look into documents that did not get annotated at all, and we find that these documents are usually of a high quality. However, their relative proportion in corpora may limit their usage.

We use a sample of the English corpus (183,497 documents, 1.3 GB) and compare the size of documents depending on the presence (or not) of annotations. The stacked counts are shown in figure 8.6.

We observe that clean document mean length is slightly shorter than non-clean ones. Also, we note that while the length standard deviation of clean documents seems to be shorter, the computation yields larger numbers, caused by outliers in the high end (Annotations: $\mu = 8606$ $\sigma = 49874$, Clean: $\mu = 6537$ $\sigma = 14983$). By removing the top and bottom 5%, we get (Annotations: $\mu = 3686$ $\sigma = 4047$, Clean: $\mu = 3582$ $\sigma = 3202$).

These results are not sufficient to state on the intrinsic quality of the clean documents, but may ease the study of the filters and identify future filtering needs.

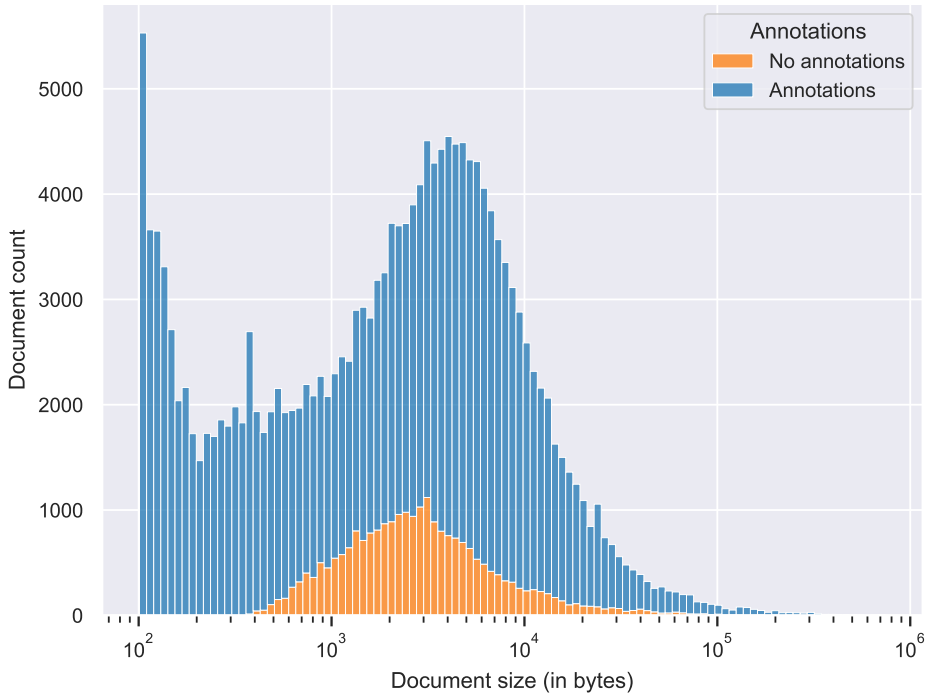


Figure 8.6: Stacked distribution of annotated and non-annotated (clean) documents on a selection of the English corpus

Adult documents

While very small in proportions, adult annotation documents highlight interesting facts.

The French sample contains 32,870 adult documents, out of 52,037,098.

We count if some documents coming from tetu.com are labeled as adult, in order to probe the possibility of finding LGBTQI+ content annotated as adult. We find 1063 documents, representing ~ 3.2% of the adult documents. This may imply that more LGBTQI+ content sites are present in the blocklist, thus increasing the ratio of LGBTQI+ content labeled as adult.

We take the first 100 adult documents of the French corpus and check whether they are properly classified.

- *true positives* documents that exhibit explicit sexual content geared towards pornography (pornographic websites, sexually explicit fictions)
- *false positives* documents that do not meet these criteria,

We separately count websites that are simultaneously non-explicit and from LGBTQI+ websites.

We find:

1. 77 true positives,
2. 2 false positives belonging to LGBTQI+ websites,
3. 21 false positives

While the majority of true positives are properly classified, numerous educational documents do appear: These type of documents exhibit an explicit language, but does feature a good document quality, and a better representation of sexuality that is less offensive compared to the usual associations between sexually explicit content and hate speech (Luccioni and Viviano, 2021).

The false positives are, for the majority, websites that do not belong in the blocklist in the first place. We assume that the addresses were previously used as adult websites.

Hard bound problems

Several pipeline steps (especially annotators), work using hard thresholds. As an example, any document that is less than 5 lines is considered to be *tiny*. However, when exploring data, we can see that there is a number of documents whose number of lines is in the neighboring of the threshold, and quality is similar to the documents labeled as *tiny*.

When plotting the distribution of clean and annotated corpus data, we can notice that a very high number of documents are of a tiny (100B) size, which coincidentally happens to be the minimum size for a document to be accepted, since the first filter removes lines that are shorter than 100 characters ($\geq$ 100B).

8.5 Discussion

8.5.1 Corpus

We provide a new, document-oriented corpus of the same size of OSCAR 21.09 that keeps document integrity and is easier to filter thanks to annotations.

While the mid and high resourced languages are of a similar size, several low resource languages have seen an important decrease of size. We still have to check whether this size decrease comes with a quality increase, since previous low resource OSCAR corpora sometimes exhibited extremely poor quality: Many non-linguistic corpora that were published and deemed unusable weeks or months after release.

We also note that documents of similar languages could have been merged into larger corpora, and we show that the German corpus holds $\sim$ 30MB of Alemannic

that, with appropriate filtering, could be treated as an independent corpus. These cases of merging are also interesting to investigate, as they can explain identification mismatches and could, in turn, help to build better language identification models. More work has to be done in order to properly map the connection between low-resource languages and mid and high resource languages potentially containing data in these languages.

8.5.2 Annotations

The selected annotations exhibit numerous caveats that have to be addressed in the future iterations of OSCAR generation pipelines.

The length-based annotations are widespread in the corpus, especially in mid to high resource languages ($\sim$ 50% in Czech) highlighting the potential low quality of a high number of documents as well as the need of better characterizing the nature of these line length discrepancies. Web crawls often contain boilerplate content extracted from headers, footers and sidebars, and these lines are present in the Common Crawl dumps. Another solution would be to base the whole OSCAR generation pipeline on raw HTML files, potentially multiplying the computational cost and complexity of generating corpora.

The *adult* annotation, based from an adult URL blocklist, is present on a very limited set of documents. However, studies have shown that adult content has been present in a previous version of OSCAR in a larger proportion than the one measured here (Kreutzer et al., 2022) (and Chapter 6), hinting at a bad performance of the blocklist based adult content filtering approach. Moreover, we noticed that the blocklist contained websites representing LGBTQI+ related topics, which damages the representation of the LGBTQI+ (association with adult content, filtering out LGBTQI+ documents, which in turn could limit the representation in downstream tasks...). Model-based approaches may help in improving the *adult* annotation, and should be the next step towards a better annotation of adult content (Luccioni and Viviano, 2021).

8.6 Conclusion

With the improvements to the Ungoliant pipeline described in this chapter and the release of OSCAR 22.01, we believe we are moving the OSCAR project in a direction were we are capable of distributing high quality up-to-date textual data for a wide range of NLP and Digital Humanities applications.

While we are aware that not all the problems and concerns around the OSCAR corpus have been addressed, we hope we can continue working on this project as it has already had a significant impact on the NLP community, especially for studies in underrepresented languages.

We believe however that the next steps in improving our corpus will require a more close involvement and participation of the OSCAR users. We thus hope that in the coming months and years we will be able to build an active open source community around the OSCAR project where people will be able to collaborate and contribute directly to the development of future versions of OSCAR and its pipeline Ungoliant.

While this chapter marks the end of the multilingual discussion of this thesis, the French sub-corpus of OSCAR will be instrumental to the development of our models and resources for both contemporary and historical French, as we will see in the coming chapters.

Part III

French Corpora

9 Contemporary French Corpora

In which we present a part of the work of Popa-Fabre et al. (2020) who construct a balanced corpus for contemporary French that could be used for language modeling; and of Ortiz Suárez et al. (2020a) who aligned both the Universal Dependencies and the TEI-annotated NER version of the French Treebank, correcting multiple annotation mistakes and discrepancies, and who then converted the NER annotations to a more machine-ready CoNLL-like format that is more often used for training neural models.[1]

Having constructed a multilingual corpus out of web data that was in theory big enough to train a state-of-the-art language model (Liu et al., 2019) for a wide range of languages, and having addressed some of the quality concerns that some researchers had expressed about this type of corpus. We wanted to focus a little more on constructing resources specifically for Contemporary and Historical French, as this was the originally intended task when we first started working on OSCAR, which was always intended to be a French corpus only, but that ended being multilingual due to the multilingual nature of Common Crawl.

In this chapter we will present *CaBeRnet* (Popa-Fabre et al., 2020) A Contemporary French Balanced Corpus that is orders of magnitude smaller than the French OSCAR sub-corpus, but that as opposed to OSCAR, it is manually curated and specifically designed to be a linguistically balanced cross-genre corpus for the French language. We will also briefly present the work of Ortiz Suárez et al. (2020a) who aligned both the Universal Dependencies and the TEI-annotated NER version of the French Treebank, giving us a more consistent a more user-friendly NER French corpus that will be used for evaluation in later chapters.

[1]Contributions: For the part of (Popa-Fabre et al., 2020) presented here, I prepared and helped with the cleaning of the Wikipedia section of CaBeRnet, I also prepared and tokenized all the corpora for the descriptive comparison. For the part of (Ortiz Suárez et al., 2020a) presented here I devised a small part of the alignment script and did some manual corrections on the NER annotations of the FTB.

9.1 Contemporary French Balanced Corpora

While working on OSCAR 2019,[2] the question of quality versus size of corpus caught our attention. We wanted to study in particular the issue of corpus "representativeness" in order to grasp to what extent a linguistically balanced cross-genre language sample would be sufficient to pre-train a language model. Here for "representativeness" we follow Biber's definition: *"representativeness refers to the extent to which a sample includes the full range of variability in a population"* (Biber, 1993).

To construct our corpora we adopt a balanced approach by sampling a wide spectrum of language use and its cross-genre variability, be it situational (e.g. format, author, addressee, purposes, settings or topics) or linguistic, e.g. linked to distributional parameters like frequencies of word classes and genres. In this fashion, we developed two corpora:

1. The French Balanced Reference Corpus (*CaBeRnet*), which includes a wide-ranging and balanced coverage of cross-genre language use to be maximally representative of the French language and therefore yield good generalizations from.

2. The *French Children Book Test* (CBT-fr), which includes both narrative material and oral language use as present in youth literature, and which could be used for domain-specific language model training.

Both corpora are inspired by existing American and English corpora, respectively COCA, the balanced Corpus of Contemporary American English (Davies, 2009, 2010), and the Children Book Test (Hill et al., 2016, CBT).

9.1.1 CaBeRnet

The CaBeRnet corpus was inspired by the genre partition of the American balanced corpus COCA,[3] which at the end of 2019, when this study was conducted, contained over 618 million words of text (20 million words each year 1990-2019) and was equally divided among spoken, fiction, popular magazines, newspapers, and academic texts (Davies, 2009, 2010). A second reference, guiding our approach and sampling method, was one of the earliest precursors of balanced reference corpora: the BNC (Consortium et al., 2007), which covered a wide variety of genres, with the intention to be a representative sample of spoken and written language.

CaBeRnet was obtained by compiling existing data-sets and web-text extracted from different sources as detailed in this subsection. As shown in Table 9.1, genres sources are evenly divided (~120 million words each) into spoken, fiction, magazine,

[2] All the work on CaBeRnet was conducted prior to the existence of OSCAR 21.09 and OSCAR 22.01. As a result, all the mentions of OSCAR in this chapter refer to OSCAR 2019.

[3] https://www.english-corpora.org/coca/

CaBeRnet Sub-set	Tokens	Unique Forms	TTR
Oral	122,864,888	291,744	0.0024
Popular	131,444,017	458,521	0.0035
News	132,708,943	462,971	0.0035
Fiction	198,343,802	983,195	0.0050
Academic	126,431,211	1,433,663	0.0113
Total	711,792,861	2,558,513	0.0036

Table 9.1: Comparison of number of unique forms in the different genres represented by CaBeRnet partition. TTR: Type-Token Ratio. Lemmatization and tokenization was performed as described in §9.1.3.

newspaper, academic to achieve genre-balanced between oral and written modality in newspapers and popular written style, technical reports and Wikipedia entries, fiction, literature and academic production.

CaBeRnet Oral The oral sub-portion gathers both oral transcriptions (ORFEO and Rhapsodie[4]) and Films subtitles (Open Subtitles.org), pruned from diacritics, interlocutors tagging and time stamps. To these transcriptions, we add the French European Parliament Proceedings (1996-2011), as presented in Koehn (2005), which contribute a sample of more complex oral style with longer sentences and richer vocabulary.

CaBeRnet Popular Press The whole sub-portion of Popular Press is gathered from an open data-set from the *Est Républicain* (1999, 2002 and 2003), a regional press format.[5] It was selected to match popular style as it is characterized by easy-to-read press style and a wide range of every-day topics characterizing local regional french press.

CaBeRnet Fiction & Literature The Fiction & Literature sub-portion was compiled from March 2019's Wiki Source and WikiBooks dump and extracted using WikiExtractor.py, a script that extracts and cleans text from a WikiMedia database dumps, by performing template expansion and preprocessing of template definitions.[6]

CaBeRnet News The News sub-portion builds upon web crawled elements, including Wikimedia's NewsComments and WikiNews reports from the May 2019 WikiMedia dump, collected with a custom version of WikiExtractor.py. We also add

[4]ORFEO corpus available at www.cocoon.huma-num.fr/exist/crdo/ ; Rhapsodie corpus at www.projet-rhapsodie.fr.
[5]Corpus available at www.cnrtl.fr/corpus/estrepublicain/.
[6]Script available at https://github.com/attardi/wikiextractor.

newspaper's content gathered by the Chambers-Rostand Corpus (i.e. Le Monde 2002-2003, La Dépêche 2002-2003, L'Humanité 2002-2003) and *Le Monde diplomatique*. This open-source corpora were assembled to represent a higher register of written news style from different political and thematic horizons. Several months of French Press Agency reports are also added (AFP, 2007-2011-2012), which contribute with a more simple and telegraphic style than the others newspaper written samples of the corpus.[7]

CaBeRnet Academic The academic genre was also built from different sources including technical and educational texts from WikiBooks and Wikipedia dump (prior to 2016) for their thematic variety of highly specialized written production. The ORFEO Corpus offered a small sample of academic writings like PhD dissertations and scientific articles encompassing a wide choice of disciplinary topics, and the TALN Corpus[8] was included to represent more concise written style characterizing scientific abstracts and proceedings.

For all sub-portions of CaBeRnet, visual inspection was performed to remove section titles, redundant meta-information linked to publishing schemes of each of the six news editor included. This was manually achieved by compiling a rich set of regular expressions specific of each textual source to obtain clean plain text as an output.

9.1.2 French Children Book Test (CBT-fr)

The French Children Book Test (CBT-fr) was built upon its original English version, the Children Book Test (CBT) (Hill et al., 2016),[9] which consists of books freely available from Project Gutenberg.[10]

Using youth literature and children books guarantees a clear narrative structure, and a large amount of dialogues, which enriches with oral register the literary style of this corpus. The English version of this corpus was originally built as a benchmark data-set to test how well language models capture meaning in context. It contains 108 books, and a vocabulary size of 53,628 tokens.

The French version of CBT, named CBT-fr, was constructed to guarantee enough linguistic similarities between the collected books in the two languages. 104 freely available books were included. One third of the books were purposely chosen because they were classical translations of English literary classics. Chapter heads, titles, notes and all types of editorial information were removed to obtain a plain

[7] This part of CaBeRnet corpus is still subject to License restrictions. However, this restricted amount of AFP news reports can reasonably fall in the public domain.

[8] TALN proceedings corpus (about 2 million) builds on a subset of 586 scientific articles (from 2007 to 2013), namely TALN and RECITAL. Available at redac.univ-tlse2.fr/corpus/taln_en.html.

[9] This data-set can be found at www.fb.ai/babi/.

[10] www.gutenberg.org.

narrative text. The effort of keeping proportion, genre, domain, and time as equal as possible yields a multilingual set of comparable corpora with a similar balance and representativeness.

Children Book Test - fr	Words
Number of different lemmas	25,139
Total number of forms	95,058
Mean number of forms per lemma	3.78
Number of lemmas having more than one form :	14,128
Percentage of lemmas with multiple forms	56.20

Table 9.2: Lexical statistics of French CBT, performed as described in §9.1.3

9.1.3 Descriptive Comparison

Having put together these two different balanced corpora, we wanted to perform a descriptive comparison between them, the French subcorpus of OSCAR 2019 (that we call OSCAR-fr for short) and Wikipedia (Wikipedia-fr). In order to perform this comparison we start by tokenizing all corpora. For this we used two different tokenizers: A standalone version of SEM (Segmenteur-Étiqueteur Markovien) (Dupont, 2017) and TreeTagger (Schmid, 1999). Both are based on cascades of regular expressions, and both perform tokenization and sentence splitting. The first was used for descriptive purposes because it technically allowed to segment and tokenize all corpora including OSCAR (23 billion words). Hence, all corpora were entirely segmented into sentences and tokenized using SEM.

While the second tokenization method was only run on 3 million words samples to automatically tag them with TreeTagger into part-of-speech and lemmatize them.[11] All corpora were randomly shuffled by sentence to then select samples of 3 million words, to be able to compare them in terms of lexical composition (Type-Token Ratio, see Table 9.4).

For Wikipedia-fr in particular we use a dump executed from April 2019, where HTML tags and tables were removed, together with template expansion using Attardi's tool WikiExtractor.[12]

Size and Composition

Length of sentences is a simple measure to quantify both sentence syntactic complexity and genre. Hence, the number of sentences reported in Table 9.3 shows interesting patterns of distributions across genres.

[11] Based on the tagset available at https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/french-tagset.html.

[12] https://github.com/attardi/wikiextractor

CORPUS	WORDFORMS	TOKENS	SENTENCES
OSCAR-fr	23,212,459,287	27,439,082,933	1,003,261,066
Wiki-fr	665,599,545	802,283,130	21,775,351
CaBeRnet	697,119,013	830,894,133	54,216,010
CBT-fr	5,697,584	6,910,201	317,239

Table 9.3: Comparing the corpora under study.

As reported on Table 9.3, in the Wikipedia-fr dataset (660 million words) sentences are relatively longer compared to other corpora. It has the advantage of having a comparable size to CaBeRnet, but its homogeneity in terms of written genre is limited to Wikipedia's entries descriptive style.

Lexical Variety

We also try to find a useful measure of complexity that measures lexical richness or variety in vocabulary. For this, we present the type-token ration (TTR) of the corpora we analyze. This measure, is generally used to assess language use aspects, like the amount of words used to communicate by language learners or children, it represents the total number of unique words (types/forms) divided by the total number of tokens in a given sample of text. Thus, the closer the TTR ratio is to 1, the greater the lexical richness of the corpus. Table 9.1 summarizes the lexical variety of the five sub-portions of CaBeRnet, respectively taken as representative of Oral, Popular, Fiction, News, and Academic genres.

Domain diversity of texts can be observed in the lexical statistics showing a gradual increase in the number of distinct lexical forms (cf. TTR). This pattern reflects a generally acknowledged distributional pattern of vocabulary-size across genres. Oral style shows a poorer lexical variety compared to newspapers/magazines' textual typology. The lexically rich fictional/classic literature is outreached by academic writing-style with its wide-ranging specialized vocabulary. All in all, Table 9.1 quantitatively suggests that the selected textual and oral materials are indeed representative of the five types of genres of CaBeRnet.

Morphological Richness

To select a measure that would help quantifying the different corpora morphological richness, we follow Bonami and Beniamine (2015). Hence, the proportion of lemmas with multiple forms in a given vocabulary size was evaluated on randomly selected samples of 3-million-words from each of the analyzed corpora (see Table 9.4).

Table 9.4 reports some more in-depth lexical and morphological statistics across corpora. Here we see that, although OSCAR is 34 times bigger than CaBeRnet, their

3M samples	CBT-fr	CaBeRnet	Wiki-fr	OSCAR-fr
nb of diff. lemmas	25,139	30,488	31,385	31,204
tot. nb forms	95,058	180,089	238,121	190,078
mean nb forms/lemma	3.78	6.19	7.85	6.40
nb lemmas > 1 form	14,128	15,927	15,182	16,480
% lemmas > 1 form	56.20	52.24	48.37	52.81

Table 9.4: Lexical statistics on morphological richness over randomly selected samples of 3 million words from each corpus. nb : number

total number of forms and the proportion of lemmas having more than one form in a 3-million-word sample are quite similar. FrWiki shows a radically different lexical distribution with numerous hapaxes but a lower morphological richness. Although its total number of forms is more than one third higher than in OSCAR and CaBeRnet samples, the proportion of lemmas having more than one distinct form is around four points below CaBeRnet and OSCAR. Comparatively, youth literature in CBT-fr shows the greatest morphological richness, with around 56% of lemmas having more than one form.

Having performed this descriptive evaluation, we will evaluate how these corpora perform as pre-training datasets for language models in the following part of the thesis. For now, we will present a small improvement that we contributed to an existing Named Entity Recognition corpus in French.

9.2 A named entity annotation layer for the UD version of the French TreeBank

As discussed in 3.3.1, Sagot et al. (2012) describe the addition to the French Treebank (FTB) (Abeillé et al., 2003) in its FTB-UC version (Candito et al., 2010) of a new, freely available annotation layer providing named entity information in terms of span and type (NER) as well as reference (NE linking), using the Wikipedia-based Aleda entity database (Sagot and Stern, 2012) as a reference entity database. This was the first freely available French corpus annotated with referential named entity information and the first freely available such corpus for the written journalistic genre. However, this annotation is provided in the form of an XML-annotated (TEI-annotated) text with sentence boundaries but no tokenization.

Since the publication of that named entity FTB annotation layer, the field has evolved in many ways. Firstly, most treebanks are now available as part of the *Universal Dependencies* (UD)[13] treebank collection (Zeman et al., 2021). Secondly, neural approaches have considerably improved the state of the art in natural language

[13] https://universaldependencies.org

processing in general and in NER in particular. In this regard, the emergence of contextual language models has played a major role. However, surprisingly few neural French NER systems have been published.[14] This might be because of the fact that getting access to the FTB with its named entity layer as well as using this corpus were not straightforward tasks.

For a number of technical reasons, re-aligning the XML-format named entity FTB annotation layer created by Sagot et al. (2012) with the *"official"* version of the FTB or, later, with the version of the FTB provided within the Universal Dependency (UD) framework was not a straightforward task.[15] Moreover, due to the intellectual property status of the source text in the FTB, the named entity annotations could only be provided to people having signed the FTB license, which prevented them from being made freely downloadable online.

Our goal in this section is to prove a new, easy-to-use UD-aligned version of the named entity annotation layer in the FTB. We describe the process whereby we re-aligned the named entity FTB annotations by Sagot et al. (2012) with the UD version of the FTB (Candito et al., 2010). This makes it possible to share these annotations in the form of a set of additional columns that can easily be pasted to the UD FTB file. This new version of the named entity FTB layer is much more readily usable than the original TEI encoded version, and will serve as a basis for our experiments in the last part of this thesis.

9.2.1 Alignment to the UD version of the FTB

The named entity (NE) annotation layer for the FTB was developed using an XML editor on the raw text of the FTB. Annotations are provided as inline XML elements within the sentence-segmented but non tokenized text. For creating our NER models, we first had to align these XML annotations with the already tokenized UD version of FTB.

Sentences were provided in the same order for both corpora, so we did not have to align them. For each sentence, we created a mapping M between the raw text of the NE-annotated FTB (i.e. after having removed all XML annotations) and tokens in the UD version of the FTB corpus. More precisely, character offsets in the FTB-NE raw text were mapped to token offsets in the tokenized FTB-UD. This alignment was done using case-insensitive character-based comparisons and were a mapping of a span in the raw text to a span in the tokenized corpus.

[14] Apart from the systems we will present in the last part of this thesis, we are only aware of the *entity-fishing* NER (and NE linking) system developed by Patrice Lopez, a freely available yet unpublished system.

[15] Note that the UD version of the FTB is freely downloadable, but does not include the original tokens or lemmas. Only people with access to the original FTB can restore this information, as required by the intellectual property status of the source text.

We used the inline XML annotations to create offline, character-level NE annotations for each sentence, and reported the NE annotations at the token level in the FTB-UD using the mapping M obtained.

We logged each error (i.e. an unaligned NE or token) and then manually corrected the corpora, as those cases were always errors in either corpora and not alignment errors. Likewise, we found 70 errors in FTB-NE and 3 errors in FTB-UD. Errors in FTB-NE were mainly XML entity problems (unhandled "&", for instance) or slightly altered text (for example, a missing comma). Errors in FTB-UD were probably the result of some XML artifacts.

9.3 Conclusion

In this chapter we have presented two balanced corpora for Contemporary French, these corpora will be used as pre-training datasets for language models in the next part of the thesis and will serve as both a baseline and a benchmark for assessing the quality of OSCAR as a pre-training corpus, at least for French.

We also presented an alignment to the UD version of the French Treebank, which can be considered as a small quality-of-life improvement that will facilitate the usage of this dataset as an evaluation of the neural models that we will train in the upcoming part of the thesis. We also believe that this new version of the NER annotated FTB will be useful to other researcher that would like to evaluate NER architectures on this dataset.

10 Historical French Corpora

In which we present a part of the work of Grobol et al. (2022) who put together a raw corpus for Medieval French intended to pre-train language models. We also present part of the work of Gabay et al. (2022) who construct a FreEM$_{max}$,a corpus for Early Modern French for the pre-training of language models, as well as FreEM$_{NER}$ an evaluation corpus annotated in named entity recognition. We also briefly present FreEM$_{LPM}$ an evaluation corpus annotated in part-of-speech tagging.[1]

Having extensively worked in Contemporary French corpora, we wanted to refocus and develop some historical French resources. As mentioned in the introduction of this thesis, this Ph.D. project was conceived and financed by ANR BASNUM (ANR-18-CE38-0003) project, whose main objective was to digitize and enrich Antoine Furetière's *Dictionnaire Universel* (DU), in its 1701 version reviewed and corrected by Basnage de Beauval (Furetière, 1701), a text written in its entirety in Early Modern French. In Section 10.2 we develop resources for Early Modern French intended for both the pre-training and the evaluation of neural language models. This will allow us to develop, in the next Part, state-of-the-art models capable of conducting the dictionary enriching task originally planned by the BASNUM project.

Having said this, we also decided to participate in the curation of a small Medieval French corpus for the pre-training of a language model. Participating in this endeavor means that at the end of this chapter we will have developed resources for French covering a period going from the 9th century to the present day, that is, we will have developed and curated textual resources for effectively all language states of French.

[1]Contributions: For the part of (Grobol et al., 2022) presented in this chapter, I just helped with the statistics presented in table 10.1 and figure 10.1. For the part (Gabay et al., 2022) presented here, I helped in the crawling of some of the Wikisource transcriptions and some transcriptions found online. For the subsection 10.2.3 I coded a script that allows to parse the corpus into a machine-readable format, find some common mistakes in annotations and generate the final split; I also made some manual corrections to the training corpus itself.

10.1 Medieval French Corpus

This section describes the raw corpus of Medieval French we gathered in order to train unsupervised language models for Old French. To our knowledge, it is one of the largest such dataset gathered for Medieval French, although it remains quite small (55 MiB in total) relatively to the corpora usually used for pre-training contextual embeddings models.

We chose to include a few texts from the early Middle French period (14^{th} -15^{th} c.) in this raw corpus, which brings a valuable complement of the prose documents that are lacking for Old French, while staying close enough to late Old French, the boundary between the two epochs being somewhat fuzzy. These texts precede the adoption of norms established by editors after the invention of Gutenberg's printing press. Middle French is more regular than Old French in some respects such as word order (Marchello-Nizia et al., 2020) and less in others such as NP structure and pronouns system (Marchello-Nizia, 1979), but they share most of their lexicon and for these relatively early texts, the syntax is not too different from that of late Old French texts.

Corpus	**Size / MiB**
BFM (Guillot et al., 2018)	20.7
AND (Rothwell et al., 2005)	17.2
NCA (Kunstmann and Stein, 2007)	9.7
Chartes Douai (Gleßgen, 2003)	3.1
OpenMedFr (Wrisley, 2018)	1.7
Geste (Jean-Baptiste-Camps et al., 2019)	1.5
MCVF (Martineau, 2008)	1.4
Chartes Aube (Reenen et al., 2007)	0.2
Total	55.3

Table 10.1: Data collection

Medieval French has many factors of variation: language evolution, dialects, domains, forms of text (verse or prose) and lack of standard. Our dataset gives us a representation of Medieval French that is as accurate and diversified as possible, given the limited amount of material that survived to these days. The detailed instructions to replicate this dataset are described in the Appendix F.1. No particular processing is done on the original documents.

In order to get a sound evaluation of the contextual embeddings trained with this dataset, we filter out the documents that are also present in the SRCMF treebank used

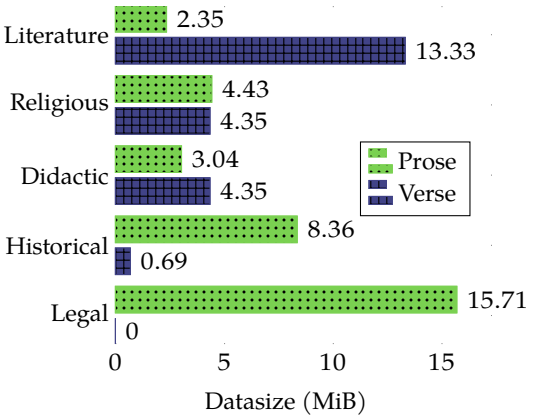


Figure 10.1: Distribution of form and domain, gathered from documents metadata and manual annotation.

for evaluation purposes in section 14.1.[2] The resulting corpus is quite heterogeneous: legal texts and verse literature are in the majority, whereas other domains, such as historical and didactic texts, are under-represented, as can be seen in fig. 10.1.

10.2 Early Modern French Corpora

For the past few years, we have been involved in the development of linguistic resources for Early Modern French. The initiative, called FreEM (which stands for *FREnch Early Modern*), aims to collect the corpora required for various NLP tasks such as lemmatization, POS tagging, linguistic normalization and named entity recognition. Two of these corpora are introduced here: FreEM$_{max}$ (see Section 10.2.1) and FreEM$_{LPM}$ (see Section 10.2.2).

10.2.1 FreEM$_{MAX}$

Usable historical documents are difficult to find because, as previously mentioned, they are more rare than contemporary ones; editors tend to normalize the language (*i.e.* use the spelling conventions of contemporary French, see (Gabay, 2014)), transcriptions are not (always) distributed in a digital format. FreEM$_{max}$ (Gabay et al., 2022) is an attempt to solve this problem, and the aim of this dataset is to group together the largest number of texts possible written in Early Modern French.

The texts we have curated have a variety of sources, which can be grouped into three main types:

[2] As noted by Gururangan et al. (2020), pre-training on task specific data provides an additional boost, this would muddle our results, since our objective here is not so much task optimization as embeddings benchmarking.

- Two institutional datasets have been used and are non open-sourced:
  - Frantext *intégral* (ATILF, 1998–b), the biggest database of French texts (only the texts between 1500 and 1800), a very small portion of which is open access: Frantext *Démonstration* (ATILF, 1998–a);
  - *Electronic Enlightenment* (Bodleian Libraries, 2008–), an online collection of edited correspondences of the Early Modern period;
- Several come from research projects distributing transcriptions online:
  - The *Antonomaz project*, French *mazarinades* (https://cahier.hypotheses.org/antonomaz);
  - The II.B section (in French) of the *Actis Pacis Westphalicae*, diplomatic letters for the Peace of Westphalia (http://kaskade.dwds.de/dstar/apwcf/);
  - The Bibliothèques virtuelles humanistes, 16th c. French literature (http://www.bvh.univ-tours.fr);
  - The *Corpus électronique de la première modernité*, 17th c. French literature (http://www.cepm.paris-sorbonne.fr)
  - The *Condé* project, *coutumiers normands* (https://conde.hypotheses.org)
  - The Corpus Descartes, works of René Descartes (https://www.unicaen.fr/puc/sources/prodescartes/);
  - The *Bibliothèque dramatique* of the CELLF, 17th c. French plays (http://bibdramatique.huma-num.fr);
  - The *Fabula numerica* project, French fables (https://obvil.sorbonne-universite.fr/projets/fabula-numerica);
  - The *Fonds Boissy*, plays of Louis de Boissy (https://www.licorn-research.fr/Boissy.html);
  - The *Mercure Galant* project, the famous French *gazette* and literary magazine between 1672 and 1710 (https://obvil.sorbonne-universite.fr/corpus/mercure-galant);
  - The *Rousseau online* project, works of Jean-Jacques Rousseau (https://www.rousseauonline.ch);
  - The *Sermo* project, sermons of the 16th and 17th c. (http://sermo.unine.ch);
  - The *Théâtre classique* project, 17th and 18th c. French plays (http://www.theatre-classique.fr);
- Additional sources come from researchers who kindly accepted to offer their personal transcriptions or data scrapped by our team:

- Transcriptions by Anne-Élisabeth Spica (17th c. French novels);
- Transcriptions found on *Wikisource* (https://fr.wikisource.org);
- Transcriptions (ePub files) found on *Gallica* (https://gallica.bnf.fr);
- Transcriptions found on various websites online.

Origin	#Tokens	Origin	#Tokens
Spica corpus	691,467	Frantext *intégral* (>1500, <1800)	60,018,390
Antonomaz project	119,194	Frantext *intégral* (>1800)	71,504,440
Acta Pacis Westphlicae II B	2,463,047	Frantext *Démonstration*	1,255,454
Bibliothèque Bleue	776,838	Gallica	5,212,333
BVH	2,434,657	Boissy project	438,215
CEPM	2,707,432	Mercure galant	5,427,469
Condé project	3,173,845	Rousseau Online project	2,428,587
Descartes	1,025,337	Scrapping	1,936,835
CELLF	1,873,772	Sermo project	529,647
Electronic enlightenment	6,568,047	Théâtre classique project	13,916,169
Fabula project	145,978	Wikisource	996,329
TOTAL			185,643,482

Table 10.2: Breakdown of the FreEM$_{max}$ corpus by text origin.

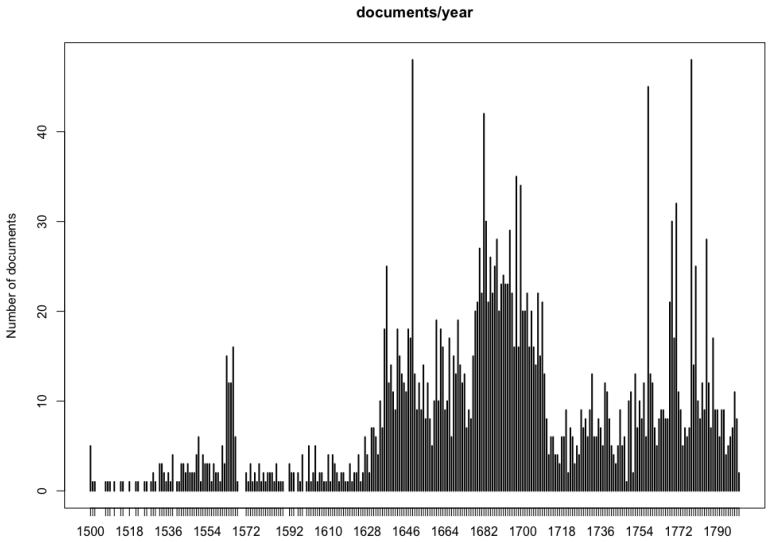


Figure 10.2: Distribution of the documents in the FreEM$_{max}$ corpus per year

Additional data for later states of the language, up to the 1920's (mainly from FRANTEXT *intégral*), are also provided for two main reasons: on the one hand, it

is common to normalize Early Modern French into Contemporary French (Gabay, 2014) because of the linguistic proximity between these the two states of the language, and on the other hand, it helps to collect (precious) additional data to avoid ending up with too small of a corpus for our needs.

The final result is far from being balanced or representative (see Figure 10.2). 16th c. French documents are under-represented, as well as 18th c. literature. The 17th c. is clearly over-represented, especially its second half—probably one of the most important of French literature, which could explain this situation (on top of our personal interest for this specific period).

As some texts are still (partially) protected by restrictive licenses, the FreEM$_{max}$ corpus exists in both open and non-open versions, only the open one being distributed.[3] In order to limit the impact of licenses forbidding the modification of files, we have designed a pipeline to distribute the data as it was found and recreate it (see Figure 10.3).

Metadata is prepared manually in order to have the same categories for each document, whatever its origin. As well as the author, the title and the date (where relevant), we also provide the genre ("theater"), sometimes a subgenre ("tragedy"), the linguistic status (normalized or not) and the license attached to the transcription.

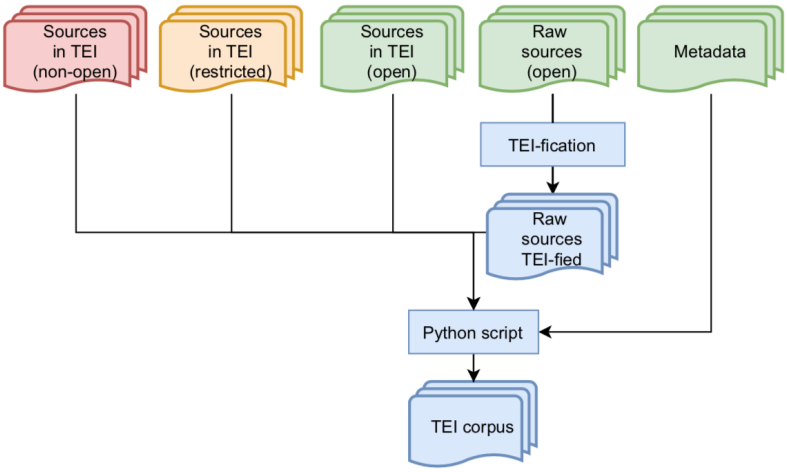


Figure 10.3: FreEM$_{max}$ compilation pipeline. All files are kept in their original format. Metadata is manually prepared in separate files in order to automatically transform and clean (in blue) all the available documents into XML TEI files following the same encoding. It allows us to distribute open data (in green) but also data distributed with restrictions regarding the modification of the original format (in orange). Non-open texts (in red) are not distributed.

[3] https://freem-corpora.github.io/corpora/max/

10.2.2 FreEM$_{LPM}$

The FreEM$_{LPM}$ (Gabay et al., 2020-10) stands for *Lemma, POS tags, Morphology*. The POS-annotated data, is a mixture of two different sources. On the one hand, there is the *CornMol* corpus (Camps et al., 2021), made up of normalized 17th c. French comedies. On the other hand, there is a gold subset of the *Presto* corpus (Blumenthal et al., 2017), made up of texts of different genres written during the 16th, 17th and 18th c., which have previously used to train annotation tools (Diwersy et al., 2017), and was heavily corrected by us to match our annotation principles (Gabay et al., 2020).

On top of traditional in-domain tests, an out-of-domain testing dataset was prepared to control the capacity of the model to generalize to other genres and periods. Centuries covered are the 16th, 17th, 18th, 19th and 20th. There are two test sets for each century: one made up only of theater, the other of everything but theater. Each test set comprises 10 short samples (c. 100 tokens), as representative as possible of the linguistic production of the century (female and male authors, decade of publication, genre, etc.).

All the data from FreEM$_{LPM}$ (but almost none of the out-of-domain) can be found in FreEM$_{max}$.

10.2.3 FreEM$_{NER}$

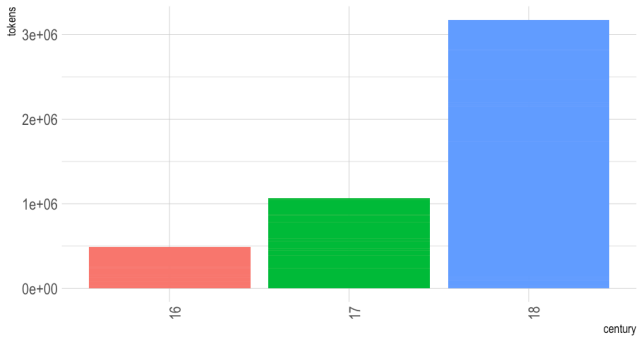


Figure 10.4: Number of tokens per century.

Rather than designing a new corpus, we have decided to use a subpart of the "core corpus" of the *Presto* project (Blumenthal et al., 2017), namely the text written during the French *Ancien Régime* (c.15th-18th c., i.e. 34 texts).[4] This choice is driven by our will to limit the number of annotated corpora for historical French, the same set of documents having already been abundantly corrected to train a lemmatizer (Gabay et al., 2020-10), but also to avoid a complex selection of works supposed to ensure a

[4]A text has been withdrawn: the *Histoire d'un voyage faict en la terre du Brésil* by Jean de Léry, the transcription being too faulty to be able to correctly annotate the document.

relative representativeness of literary documents from the *Ancien Régime*, already perfectly done by our colleagues.

The number of genres covered is very large: poetry, drama, novel, correspondence, grammar, philosophy, short stories, encyclopedic literature, etc. and guarantees, here again, a reasonable representativeness of the range of possibilities of *Belles-Lettres*.[5] The corpus is balanced regarding the distribution per century (approx 10/century) but not regarding the length of the texts, which increases over time (cf. fig. 10.4), following a possible trend in literature.

Annotation

Person		Function			
pers.ind	pers.coll	func.ind	func.coll		
Location			**Production**		
loc.adm.town	loc.phys.geo	loc.fac	prod.art	prod.rule	prod.object
loc.adm.reg	loc.phys.hydro	loc.oro			
loc.adm.nat					
loc.adm.sup					
Organization		**Time**	**Event**	**Quantity**	
org.adm	org.ent	time.date.abs	event	amount	
		time.date.rel			

Table 10.3: Types (in gray) and subtypes taken from the *Quaero* typology.

Because two important historical corpora presented *supra* (*Quaero* and *Impresso*) have chosen to follow the *Quaero* annotation guide (Rosset et al., 2011), it seemed logical to use this same typology. Because our texts and interests diverge from those of the aforementioned corpora, only some types and subtypes have been kept (cf. tab. 10.3) from the *Quaero* annotation scheme. The details of our annotation choices can be found in a dedicated annotation manual (Gabay et al., 2020).

Token	Lemma	POS	COARSE	FINE	FINE-COMP	NESTED	Wikidata ID
Les	le	Da	O	O	O	O	_
allemands	allemand	Nc	O	O	O	O	_
élurent	élire	Vvc	O	O	O	O	_
pour	pour	S	O	O	O	O	_
empereur	empereur	Nc	B-pers	B-pers.ind	B-comp.title	O	Q438435
Rodolphe	Rodolphe	Np	I-pers	I-pers.ind	B-comp.name	O	Q438435
duc	duc	Nc	I-pers	I-pers.ind	B-comp.title	O	Q438435
de	de	S	I-pers	I-pers.ind	I-comp.title	O	Q438435
Suabe	Souabe	Np	I-pers	I-pers.ind	I-comp.title	B-loc.adm.reg	Q438435

Table 10.4: NERC Fine-Grained annotation avec EL

[5] We do not offer a detailed description of the genres covered, these overlapping easily: poetry can be theological, political correspondence, etc.

The annotated texts are available in multi-columns `tsv` files (cf. tab. 10.4). Each token has a lemma (manually corrected) and a POS (produced by the *Presto* project, non-systematically corrected but fairly reliable) using the MULTEXT tag set. We propose a coarse-grained annotation for high-level entity types and fine-grained annotation using subtypes using the following syntax:

`BIO-TYPE.SUBTYPE`
For instance: `B-loc.adm.town`

Subtypes are sometimes simple (`B-org.town`) sometimes double (`B-loc.phys.geo`), depending of the complexity of the entity to annotate. Nested entities (*i.e.* an entity in an entity, such as a place name in a person name in *Henri d'**Angleterre***, "Henry of England") follow exactly the same syntax, and components a similar one, using six transverse elements:

- `name` to annotate tokens that are names (*Louis*, *Philippe*...)
- `title` to annotate tokens that are titles (*sieur*, *duc*, *abbé*...)
- `qualifier` to annotate tokens that are adjectives (*l'Inde **orientale***, *l'Arabie **heureuse***, *la mer **athlantique***, *l'**ancienne** Colchide*)... but also the generation (*Henri **IV***) or a cardinal position
- `kind` to annotate tokens that are hyperonyms (*l'**Empire** de Constantinople*, *la **mer** du Japon*
- `unit` to annotate tokens that are units (meters, league, inches, pounds...)
- `val` to annotate tokens that are values (a number) that is linked to a unit to annotate an `amount`.

We have decided not to annotate metaphorical uses differently or in a separate column: everything is annotated in a literal sense. Thus, in ***France** goes to war*, *France* is labelled `loc.adm.nat` (*i.e.* the country) and not `org.adm` (*i.e.* the French government).

We have also started a first phase of semantic annotation, using Wikidata (Vrandečić and Krötzsch, 2014-09-23) identifiers, which remains imperfect. Due to the complexity of analyzing certain entities, in particular personal names (e.g. *Pope John*), it was decided to annotate them only very marginally, only in the event of the absence of ambiguity (e.g. *Pope John V*). The annotation of place names, on the other hand, is more advanced and almost exhaustive.

A first layer of annotation was made using regular expressions, before moving on to a manual correction phase. Given the size of the corpus, it is obvious that each token has not been checked, and that the final result does not claim to be perfect. Occasional checks, however, concluded that the annotation was of high enough

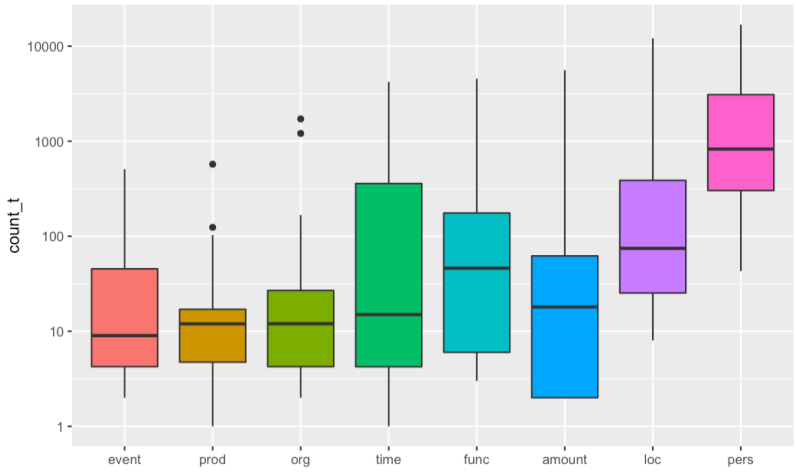


Figure 10.5: Number of entities ($\log_{10}$ scale) per category.

quality to move on to the training phase. All the annotation work was carried out by a single person, in order to ensure the consistency of the data. The structure of the file and the form of the tags was controlled by a specific parser, designed specifically for this corpus. Figure 10.5 shows the distribution of the coarse entity categories throughout FreEM$_{NER}$ on a logarithmic scale. For more detail please refer to figures G.1 and G.2 in the appendix.

10.3 Conclusion

In this chapter we have presented two raw textual corpora for historical French intended to be used in the pre-training of state-of-the-art language models, one for Medieval French and another one for Early Modern French. These corpora will be used in the next par of this thesis to produce two language models in order to tackle the textual enriching task proposed by the BASNUM project. As these two corpora are in fact quite general and diverse, we believe that the models they will produce will allow researcher in Digital Humanities to enrich and better study not only for the *Dictionnaire Universel*, but also any other text in Early Modern or Medieval French.

We have also presented a NER annotated corpus in Early Modern French, that will allow not only to evaluate our language models for Early Modern French in the upcoming part of the thesis, but also to produce a general ready-to-use state-of-the-art model for NER in Early Modern French.

Part IV

Models and Evaluation

11 CamemBERT

In which we present a part of the work of Martin et al. (2020) who pre-trained the first transformer based language model for Contemporary French using the French subcorpus of OSCAR 2019 (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020b). The model that we call CamemBERT is then evaluated in dependency parsing, part-of-speech tagging, named entity recognition and natural language inference. We also study the question of how corpus size and diversity affects the performance of an architecture like RoBERTa (Liu et al., 2019) in downstream tasks.[1]

Having extensively worked into creating and curating textual resources in previous chapters and parts of this thesis, we wanted to use these resources in order to train a monolingual contextual language model for Contemporary French.

When we started the experiments that will be discussed in this chapter, the availability of large monolingual transformer based models was limited to English-only models (Devlin et al., 2019; Radford et al., 2019; Liu et al., 2019; Yang et al., 2019; Raffel et al., 2020) and most of the work in other languages was being done through multilingual models like mBERT (Devlin et al., 2019). And even though multilingual models gave remarkable results at the time, they were often larger, and their results, as we will observe for French, could lag behind their monolingual counterparts for high-resource languages.

In order to reproduce and validate results that had so far only been obtained for English, we took advantage of the first version of OSCAR[2] (Ortiz Suárez et al., 2019) which had just been released at that time. We used the French subcorpus of OSCAR

[1] Contributions: I prepared OSCAR 2019 for the pre-training of CamemBERT and actually had to re-write the whole pipeline in order to produce the first unshuffled version of OSCAR. I did all the experiments where CamemBERT is used in embedding form. I also wrote the code to synchronize and extract fixed token embeddings from CamemBERT which was necessary at the time since this option did not exist in Hugging Face Transformer library at the time. Moreover, the whole section 11.3 and one of the main scientific contributions of the article was originally devised by me as one of the experiments that we wanted to conduct for the OSCAR project and was supposed to be part of (Ortiz Suárez et al., 2020b) presented in chapter 5. However, due to time and space constraints we preferred to do these experiments as part of the CamemBERT project. Finally, I actively participated in writing (Martin et al., 2020).

[2] Now OSCAR 2019.

2019 to train a monolingual language model for French, dubbed CamemBERT. We also trained alternative versions of CamemBERT on different smaller corpora with different levels of homogeneity in genre and style in order to assess the impact of these parameters on downstream task performance. CamemBERT used the RoBERTa architecture (Liu et al., 2019).

We then evaluated our model on four different downstream tasks for French: part-of-speech (POS) tagging, dependency parsing, named entity recognition (NER) and natural language inference (NLI). CamemBERT improved on the state of the art in all four tasks compared to previous monolingual and multilingual approaches including mBERT, XLM and XLM-R, which confirmed the effectiveness of pre-trained contextual language models for French.

11.1 CamemBERT: A Contemporary French Language Model

In this section, we describe the pre-training data, architecture, training objective and optimization setup we use for CamemBERT.

11.1.1 Training data

Pre-trained language models benefits from being trained on large datasets (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020). We therefore use the French subcorpus of OSCAR 2019 (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020b). No other filtering is done. We use the deduplicated non-shuffled version of the French subcorpus, which amounts to 138GB of raw text and to around 32.7B tokens after subword tokenization.

11.1.2 Pre-processing

We segment the input text data into subword units using SentencePiece (Kudo and Richardson, 2018). SentencePiece is an extension of Byte-Pair encoding (BPE) (Sennrich et al., 2016) and WordPiece (Kudo, 2018) that does not require pre-tokenization (at the word or token level), thus removing the need for language-specific tokenisers. We use a vocabulary size of 32k subword tokens. These subwords are learned on 10^7 sentences sampled randomly from the pre-training dataset. We do not use subword regularization (i.e. sampling from multiple possible segmentations) for the sake of simplicity.

11.1.3 Language Modeling

Transformer Similar to RoBERTa and BERT, CamemBERT is a multi-layer bidirectional Transformer (Vaswani et al., 2017). CamemBERT uses the original architectures of $\text{BERT}_{\text{BASE}}$ (12 layers, 768 hidden dimensions, 12 attention heads, 110M

parameters) and $\text{BERT}_{\text{LARGE}}$ (24 layers, 1024 hidden dimensions, 16 attention heads, 335M parameters). CamemBERT is very similar to RoBERTa, the main difference being the use of whole-word masking and the usage of SentencePiece tokenization (Kudo and Richardson, 2018) instead of WordPiece (Schuster and Nakajima, 2012).

Pretraining Objective We train our model on the Masked Language Modeling (MLM) task. Given an input text sequence composed of N tokens $x_1, ..., x_N$, we select 15% of tokens for possible replacement. Among those selected tokens, 80% are replaced with the special `<MASK>` token, 10% are left unchanged and 10% are replaced by a random token. The model is then trained to predict the initial masked tokens using cross-entropy loss.

Following the RoBERTa approach, we dynamically mask tokens instead of fixing them statically for the whole dataset during preprocessing. This improves variability and makes the model more robust when training for multiple epochs.

Since we use SentencePiece to tokenize our corpus, the input tokens to the model are a mix of whole words and subwords. An upgraded version of BERT[3] and Joshi et al. (2020) have shown that masking whole words instead of individual subwords leads to improved performance. Whole-word Masking (WWM) makes the training task more difficult because the model has to predict a whole word rather than predicting only part of the word given the rest. We train our models using WWM by using white spaces in the initial non-tokenized text as word delimiters.

WWM is implemented by first randomly sampling 15% of the words in the sequence and then considering all subword tokens in each of this 15% for candidate replacement. This amounts to a proportion of selected tokens that is close to the original 15%. These tokens are then either replaced by `<MASK>` tokens (80%), left unchanged (10%) or replaced by a random token.

Subsequent work has shown that the next sentence prediction (NSP) task originally used in BERT does not improve downstream task performance (Conneau and Lample, 2019; Liu et al., 2019), thus we also remove it.

Optimization Following (Liu et al., 2019), we optimize the model using Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.98$) for 100k steps with large batch sizes of 8192 sequences, each sequence containing at most 512 tokens. We enforce each sequence to only contain complete paragraphs (which correspond to lines in the pre-training dataset).

Pre-training We use the RoBERTa implementation in the fairseq library (Ott et al., 2019). Our learning rate is warmed up for 10k steps up to a peak value of 0.0007 instead of the original 0.0001 given our large batch size, and then fades to zero with polynomial decay. Unless otherwise specified, our models use the BASE architecture,

[3] https://github.com/google-research/bert/blob/master/README.md

and are pre-trained for 100k backpropagation steps on 256 Nvidia V100 GPUs (32 GB each) for a day. We do not train our models for longer due to practical considerations, even though the performance still seemed to continue increasing afterwards.

11.1.4 Using CamemBERT for downstream tasks

We use the pretrained CamemBERT in two ways. In the first one, which we refer to as *fine-tuning*, we fine-tune the model on a specific task in an end-to-end manner. In the second one, referred to as *feature-based embeddings* or simply *embeddings*, we extract frozen contextual embedding vectors from CamemBERT. These two complementary approaches shed light on the quality of the pretrained hidden representations captured by CamemBERT.

Fine-tuning For each task, we append the relevant predictive layer on top of CamemBERT's architecture. Following the work done on the BERT paper (Devlin et al., 2019), for sequence tagging and sequence labeling we append a linear layer that respectively takes as input the last hidden representation of the `<s>` special token and the last hidden representation of the first subword token of each word. For dependency parsing, we plug a bi-affine graph predictor head as inspired by Dozat and Manning (2017). We fine-tune on XNLI by adding a classification head composed of one hidden layer with a non-linearity and one linear projection layer, with input dropout for both.

We fine-tune CamemBERT independently for each task and each dataset, optimizing the model using the Adam optimizer (Kingma and Ba, 2015) with a fixed learning rate. Likewise, we run a grid search on a combination of learning rates and batch sizes. Furthermore, we select the best model on the validation set out of the 30 first epochs. For NLI we use the default hyper-parameters provided by the authors of RoBERTa on the MNLI task.[4] Although this might have pushed the performances even further, we do not apply any regularization techniques such as weight decay, learning rate warm-up or discriminative fine-tuning, except for NLI. We show that fine-tuning CamemBERT in a straightforward manner leads to state-of-the-art results on all tasks and outperforms the existing multilingual BERT-based models in all cases. The POS tagging, dependency parsing, and NER experiments are run using Hugging Face's Transformer library extended to support CamemBERT and dependency parsing (Wolf et al., 2019). The NLI experiments use the fairseq library following the RoBERTa implementation.

Embeddings Following Straková et al. (2019) and Straka et al. (2019) for mBERT and the English BERT, we make use of CamemBERT in a feature-based embeddings

[4] More details at https://github.com/pytorch/fairseq/blob/master/examples/roberta/README.glue.md.

setting. In order to obtain a representation for a given token, we first compute the average of each sub-word's representations in the last four layers of the Transformer, and then average the resulting sub-word vectors.

We evaluate CamemBERT in the embeddings setting for POS tagging, dependency parsing and NER; using the open-source implementations of Straka et al. (2019) and Straková et al. (2019).[5]

Dowstream Tasks For POS tagging and dependency parsing, we run our experiments using the Universal Dependencies (UD)[6] framework and its corresponding UD POS tag set (Petrov et al., 2012) and UD treebank collection (Nivre et al., 2018), which was used for the CoNLL 2018 shared task (Seker et al., 2018). We perform our evaluations on the four freely available French UD treebanks in UD v2.2: GSD (McDonald et al., 2013), Sequoia[7] (Candito and Seddah, 2012; Candito et al., 2014), Spoken (Lacheret et al., 2014; Bawden et al., 2014),[8] and ParTUT (Sanguinetti and Bosco, 2015).

For NER, we use the French Treebank (FTB) (Abeillé et al., 2003) in its 2008 version introduced by Candito and Crabbé (2009) and with NER annotations by Sagot et al. (2012). More precisely, we used the corrected and synchronized version (Ortiz Suárez et al., 2020a) presented in subsection 9.2.1.

Finally, we evaluate our model on NLI, using the French part of the XNLI dataset (Conneau et al., 2018). The XNLI dataset is the extension of the Multi-Genre NLI (MultiNLI) corpus (Williams et al., 2018) to 15 languages by translating the validation and test sets manually into each of those languages. The English training set is machine translated for all languages other than English.

11.2 Evaluation of CamemBERT

In this section, we measure the performance of our models by evaluating them on the four aforementioned tasks: POS tagging, dependency parsing, NER and NLI.

POS Tagging and Dependency Parsing For POS tagging and dependency parsing, we compare CamemBERT with other models in the two settings: *fine-tuning* and as *feature-based embeddings*. We report the results in Table 11.1.

CamemBERT reaches state-of-the-art scores on all treebanks and metrics in both scenarios. The two approaches achieve similar scores, with a slight advantage for

[5] UDPipe Future is available at `https://github.com/CoNLL-UD-2018/UDPipe-Future`, and the code for nested NER is available at `https://github.com/ufal/acl2019_nested_ner`.
[6] `https://universaldependencies.org`.
[7] `https://deep-sequoia.inria.fr`.
[8] Speech transcript uncased that includes annotated disfluencies without punctuation.

Model	GSD		Sequoia		Spoken		ParTUT	
	UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS
mBERT (fine-tuned)	97.48	89.73	98.41	91.24	96.02	78.63	97.35	91.37
$XLM_{MLM-TLM}$ (fine-tuned)	98.13	90.03	98.51	91.62	96.18	80.89	97.39	89.43
UDify (Kondratyuk and Straka, 2019)	97.83	91.45	97.89	90.05	96.23	80.01	96.12	88.06
UDPipe Future (Straka, 2018)	97.63	88.06	98.79	90.73	95.91	77.53	96.93	89.63
+ mBERT + Flair (emb.) (Straka et al., 2019)	97.98	90.31	**99.32**	93.81	**97.23**	81.40	97.64	92.47
CamemBERT (fine-tuned)	**98.18**	**92.57**	99.29	**94.20**	96.99	81.37	**97.65**	**93.43**
UDPipe Future + CamemBERT (embeddings)	97.96	90.57	99.25	93.89	97.09	**81.81**	97.50	92.32

Table 11.1: **POS** and **dependency parsing** scores on 4 French treebanks, reported on test sets assuming gold tokenization and segmentation (best model selected on validation out of 4). Best scores in bold, second best underlined.

the fine-tuned version of CamemBERT, thus questioning the need for complex task-specific architectures such as UDPipe Future.

Despite a much simpler optimization process and no task specific architecture, fine-tuning CamemBERT outperforms UDify on all treebanks and sometimes by a large margin (e.g. +4.15% LAS on Sequoia and +5.37 LAS on ParTUT). CamemBERT also reaches better performance than other multilingual pre-trained models such as mBERT and $XLM_{MLM-TLM}$ on all treebanks.

CamemBERT achieves overall slightly better results than the previous state-of-the-art and task-specific architecture UDPipe Future+mBERT+Flair, except for POS tagging on Sequoia and POS tagging on Spoken, where CamemBERT lags by 0.03% and 0.14% UPOS respectively. UDPipe Future+mBERT+Flair uses the contextualized string embeddings Flair (Akbik et al., 2018), which are in fact pre-trained contextualized character-level word embeddings specifically designed to handle misspelled words as well as subword structures such as prefixes and suffixes. This design choice might explain the difference in score for POS tagging with CamemBERT, especially for the Spoken treebank where words are not capitalized, a factor that might pose a problem for CamemBERT which was trained on capitalized data, but that might be properly handle by Flair on the UDPipe Future+mBERT+Flair model.

Named-Entity Recognition For NER, we similarly evaluate CamemBERT in the fine-tuning setting and as input embeddings to the task specific architecture LSTM+CRF. We report these scores in Table 11.2.

In both scenarios, CamemBERT achieves higher F1 scores than the traditional CRF-based architectures (both non-neural and neural), and than the fine-tuned multilingual BERT models.[9]

Using CamemBERT as embeddings to the traditional LSTM+CRF architecture gives slightly higher scores than by fine-tuning the model (89.08 vs. 89.55). This

[9] $XLM_{MLM-TLM}$ is a lower-case model. Case is crucial for NER, therefore we do not report its low performance (84.37%)

Model	F1
SEM (CRF) (Dupont, 2017)	85.02
LSTM-CRF (Dupont, 2017)	85.57
mBERT (fine-tuned)	87.35
CamemBERT (fine-tuned)	<u>89.08</u>
LSTM+CRF+CamemBERT (embeddings)	**89.55**

Table 11.2: **NER** scores on the FTB (best model selected on validation out of 4). Best scores in bold, second best underlined.

demonstrates that even though CamemBERT can be used successfully without any task-specific architecture, it can still produce high quality contextualized embeddings that might be useful in scenarios where powerful downstream architectures exist.

Natural Language Inference On the XNLI benchmark, we compare CamemBERT to previous state-of-the-art multilingual models in the fine-tuning setting. In addition to the standard CamemBERT model with a BASE architecture, we train another model with the LARGE architecture, referred to as $\text{CamemBERT}_{\text{LARGE}}$, for a fair comparison with $\text{XLM-R}_{\text{LARGE}}$. This model was trained with the CCNet corpus, described in Sec. 11.3, for 100k steps.[10] We expect that training the model for longer would yield even better performance.

Model	Acc.	#Params
mBERT (Devlin et al., 2019)	76.9	175M
$\text{XLM}_{\text{MLM-TLM}}$ (Conneau and Lample, 2019)	<u>80.2</u>	250M
$\text{XLM-R}_{\text{BASE}}$ (Conneau et al., 2020)	80.1	270M
CamemBERT (fine-tuned)	**82.5**	110M
Supplement: LARGE models		
$\text{XLM-R}_{\text{LARGE}}$ (Conneau et al., 2020)	<u>85.2</u>	550M
$\text{CamemBERT}_{\text{LARGE}}$ (fine-tuned)	**85.7**	335M

Table 11.3: **NLI** accuracy on the French XNLI test set (best model selected on validation out of 10). Best scores in bold, second best underlined.

CamemBERT reaches higher accuracy than its BASE counterparts reaching +5.6% over mBERT, +2.3 over $\text{XLM}_{\text{MLM-TLM}}$, and +2.4 over $\text{XLM-R}_{\text{BASE}}$. CamemBERT also uses as few as half as many parameters (110M vs. 270M for $\text{XLM-R}_{\text{BASE}}$).

$\text{CamemBERT}_{\text{LARGE}}$ achieves a state-of-the-art accuracy of 85.7% on the XNLI benchmark, as opposed to 85.2, for the recent $\text{XLM-R}_{\text{LARGE}}$.

[10] We train our LARGE model with the CCNet corpus for practical reasons, mainly due to the fact that it was more readily available on the Facebook infrastructure we used to train CamemBERT. Given that BASE models reach similar performance when using OSCAR or CCNet as pretraining corpus (Appendix Table E.2), we expect an OSCAR LARGE model to reach comparable scores.

CamemBERT uses fewer parameters than multilingual models, mostly because of its smaller vocabulary size (e.g. 32k vs. 250k for XLM-R). Two elements might explain the better performance of CamemBERT over XLM-R. Even though XLM-R was trained on an impressive amount of data (2.5TB), only 57GB of this data is in French, whereas we used 138GB of French data. Additionally, XLM-R also handles 100 languages, and the authors show that when reducing the number of languages to 7, they can reach 82.5% accuracy for French XNLI with their BASE architecture.

Summary of CamemBERT's results CamemBERT improves the state of the art for the 4 downstream tasks considered, thereby confirming the usefulness of a monolingual Transformer-based models for contemporary French. We obtain these results when using CamemBERT as a fine-tuned model or when used as contextual embeddings with task-specific architectures. This questions the need for more complex downstream architectures, similar to what was shown for English (Devlin et al., 2019). Additionally, this suggests that CamemBERT is also able to produce high-quality representations out-of-the-box without further tuning.

11.3 Impact of corpus origin and size

In this section we investigate the influence of the homogeneity and size of the pre-training corpus on downstream task performance. With this aim, we train alternative version of CamemBERT by varying the pre-training datasets. For this experiment, we fix the number of pre-training steps to 100k, and allow the number of epochs to vary accordingly (more epochs for smaller dataset sizes). All models use the BASE architecture.

In order to investigate the need for homogeneous clean data versus more diverse and possibly noisier data, we use alternative sources of pre-training data in addition to OSCAR 2019:

- **Wikipedia**, which is homogeneous in terms of genre and style. We use the official 2019 French Wikipedia dumps.[11] We remove HTML tags and tables using Giuseppe Attardi's *WikiExtractor*.[12]

- **CCNet** (Wenzek et al., 2020), a dataset extracted from Common Crawl with a different filtering process than for OSCAR. It was built using a language model trained on Wikipedia, in order to filter out bad quality texts such as code or tables.[13] As this filtering step biases the noisy data from Common Crawl to more Wikipedia-like text, we expect CCNet to act as a middle ground between the unfiltered "noisy" OSCAR 2019 dataset, and the "clean" Wikipedia

[11] https://dumps.wikimedia.org/backup-index.html.
[12] https://github.com/attardi/wikiextractor.
[13] We use the HEAD split, which corresponds to the top 33% of documents in terms of filtering perplexity.

dataset. As a result of the different filtering processes, CCNet contains longer documents on average compared to OSCAR 2019 with smaller—and often noisier—documents weeded out.

Table 11.4 summarizes statistics of these different corpora.

Corpus	Size	#tokens	#docs	Tokens/doc Percentiles: 5%	50%	95%
Wikipedia	4 GB	990M	1.4M	102	363	2530
CCNet	135 GB	31.9B	33.1M	128	414	2869
OSCAR 2019	138 GB	32.7B	59.4M	28	201	1946

Table 11.4: Statistics on the pre-training datasets used.

In order to make a fair comparison between these three sources of pre-training data, we randomly sample 4 GB (the size of Wikipedia) of text (at the document level) from OSCAR and CCNet, thereby creating samples of both Common-Crawl-based corpora of the same size as the French Wikipedia. These smaller 4GB samples also provides us a way to investigate the impact of pre-training data size. Downstream task performance for our alternative versions of CamemBERT are provided in Table 11.5. The upper section reports scores in the fine-tuning setting while the lower section reports scores for the embeddings.

11.3.1 Common Crawl vs. Wikipedia?

Table 11.5 clearly shows that models trained on the 4 GB versions of OSCAR 2019 and CCNet (Common Crawl) perform consistently better than the one trained on the French Wikipedia. This is true both in the fine-tuning and embeddings setting. Unsurprisingly, the gap is larger on tasks involving texts whose genre and style are more divergent from those of Wikipedia, such as tagging and parsing on the Spoken treebank. The performance gap is also very large on the XNLI task, probably as a consequence of the larger diversity of Common-Crawl-based corpora in terms of genres and topics. XNLI is indeed based on multiNLI which covers a range of genres of spoken and written text.

The downstream task performances of the models trained on the 4 GB version of CCNet and OSCAR are much more similar.[14]

[14]We provide the results of a model trained on the whole CCNet corpus in the Appendix. The conclusions are similar when comparing models trained on the full corpora: downstream results are similar when using OSCAR or CCNet.

Dataset	Size	GSD		Sequoia		Spoken		ParTUT		Average		NER	NLI
		UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS	F1	Acc.
Fine-tuning													
Wiki	4GB	98.28	93.04	98.74	92.71	96.61	79.61	96.20	89.67	97.45	88.75	89.86	78.32
CCNet	4GB	98.34	93.43	98.95	93.67	96.92	**82.09**	96.50	**90.98**	97.67	**90.04**	90.46	**82.06**
OSCAR	4GB	98.35	93.55	98.97	93.70	96.94	81.97	96.58	90.28	97.71	89.87	90.65	81.88
OSCAR	138GB	**98.39**	**93.80**	**98.99**	**94.00**	**97.17**	81.18	**96.63**	90.56	**97.79**	89.88	**91.55**	81.55
Embeddings (with UDPipe Future (tagging, parsing) or LSTM+CRF (NER))													
Wiki	4GB	98.09	92.31	98.74	93.55	96.24	78.91	95.78	89.79	97.21	88.64	91.23	-
CCNet	4GB	**98.22**	**92.93**	99.12	94.65	97.17	**82.61**	**96.74**	89.95	97.81	90.04	**92.30**	-
OSCAR	4GB	98.21	92.77	99.12	**94.92**	97.20	82.47	**96.74**	**90.05**	**97.82**	**90.05**	91.90	-
OSCAR	138GB	98.18	92.77	**99.14**	94.24	**97.26**	82.44	96.52	89.89	97.77	89.84	91.83	-

Table 11.5: Results on the four tasks using language models pre-trained on data sets of varying homogeneity and size, reported on validation sets (average of 4 runs for POS tagging, parsing and NER, average of 10 runs for NLI).

11.3.2 How much data do you need?

An unexpected outcome of our experiments is that the model trained "only" on the 4 GB sample of OSCAR 2019 performs remarkably similarly to the standard CamemBERT trained on the whole 138 GB OSCAR 2019. The only task with a large performance gap is NER, where "138 GB" models are better by 0.9 F1 points. This could be due to the higher number of named entities present in the larger corpora, which is beneficial for this task. On the contrary, other tasks don't seem to gain from the additional data.

In other words, when trained on corpora such as OSCAR and CCNet, which are heterogeneous in terms of genre and style, 4 GB of uncompressed text is large enough as pre-training corpus to reach state-of-the-art results with the BASE architecture, better than those obtained with mBERT (pre-trained on 60 GB of text).[15] This calls into question the need to use a very large corpus such as OSCAR or CCNet when training a monolingual Transformer-based language model such as BERT or RoBERTa. Not only does this mean that the computational (and therefore environmental) cost of training a state-of-the-art language model can be reduced, but it also means that CamemBERT-like models can be trained for all languages for which a CommonCrawl-based corpus of 4 GB or more can be created. OSCAR is available in more than 150 languages, and provides such a corpus for around 38 languages. Moreover, it is possible that slightly smaller corpora (e.g. down to 1 GB) could also prove sufficient to train high-performing language models. We obtained our results with BASE architectures. Further research is needed to confirm the validity of our findings on larger architectures and other more complex natural language understanding tasks. However, even with a BASE architecture and 4 GB of training data, the validation loss is still decreasing beyond 100k steps (and 400 epochs). This suggests that we

[15] The OSCAR-4 GB model gets slightly better XNLI accuracy than the full OSCAR-138 GB model (81.88 vs. 81.55). This might be due to the random seed used for pre-training, as each model is pre-trained only once.

are still under-fitting the 4 GB pre-training dataset, training longer might increase downstream performance.

11.4 Discussion

Between the pre-publication of this work[16] and the publication of its peer-reviewed version (Martin et al., 2020), many monolingual language models appeared, e.g. (Le et al., 2020b; Virtanen et al., 2019; Delobelle et al., 2020), and for as much as 30 languages (Nozza et al., 2020). In almost all tested configurations they displayed better results than multilingual language models such as mBERT (Pires et al., 2019). Interestingly, Le et al. (2020b) showed that using FlauBERT, another RoBERTa-based language model for Contemporary French, which was trained on less but more edited data, in conjunction to CamemBERT in an ensemble system could improve the performance of a parsing model and establish a new state-of-the-art in constituency parsing for Contemporary French, highlighting thus the complementarity of both models.[17]

As it was the case for English when BERT was first released, the availability of similar scale language models for Contemporary French enabled interesting applications, such as large scale anonymization of legal texts, where CamemBERT-based models established a new state-of-the-art on this task (Benesty, 2019), or the first large question answering experiments on a French Squad data set that was released after the publication of CamemBERT (d'Hoffschmidt et al., 2020) where the authors matched human performance using $\text{CamemBERT}_{\text{LARGE}}$. Being the first pre-trained Trasnformer-based language model that used the OSCAR corpus and given its impact on the community, CamemBERT paved the way for many works on monolingual language models that followed. Furthermore, the availability of all its training data favors reproducibility and is a step towards better understanding such models and the impact that the pre-training data has on them. In that spirit, we make the models used in our experiments available via our website[18] and via the `huggingface` and `fairseq` APIs, in addition to the base CamemBERT model.

11.5 Conclusion

In this chapter we investigated the feasibility of training a Transformer-based language model for languages other than Contemporary English. Using Contemporary French as an example, we trained CamemBERT, a language model based on RoBERTa. We evaluated CamemBERT on four downstream tasks (part-of-speech tagging, dependency parsing, named entity recognition and natural language inference) in

[16] https://arxiv.org/abs/1911.03894v1 (First ArXiv version).
[17] We refer the reader to (Le et al., 2020b) for a comprehensive benchmark and details therein.
[18] https://camembert-model.fr

which our best model reached or improved the state of the art in all tasks considered, even when compared to strong multilingual models such as mBERT, XLM and XLM-R, while also having fewer parameters.

Our experiments confirm the previous findings presented in chapter 5 that using web crawled data with high variability is preferable to using Wikipedia-based data. In addition, we showed that our models could reach surprisingly high performances with as low as 4 GB of pre-training data, questioning thus the need for large scale pre-training corpora. This shows that state-of-the-art Transformer-based language models can be trained on languages with far fewer resources than previously believed, and whenever a few gigabytes of data are available. This paves the way for the rise of monolingual contextual pre-trained language models for mid- and low-resourced languages. The question of knowing whether pre-training on small domain specific content will be a better option than transfer learning techniques such as fine-tuning remains open, and we will partially study it in the context of historical data in upcoming chapters.

12 FrELMo

In which we present a part of the work of Ortiz Suárez et al. (2020a) who pre-train an ELMo model for Contemporary French and then evaluate its performance in the NER annotated FTB against all the available versions of CamemBERT. From these experiments Ortiz Suárez et al. (2020a) set a new state of the art for this corpus. We also present part of the work of Popa-Fabre et al. (2020) who further train pre-train ELMo models with the previously presented CaBeRnet and CBT-fr and then evaluate them in multiple downstream tasks in order to assess the importance of representative and balanced corpora as pre-training datasets.[1]

Having trained the RoBERTa (Liu et al., 2019) based CamemBERT (Martin et al., 2020) models in the previous chapter, we wanted to fairly compare the Transformer-based architecture with ELMo (Peters et al., 2018), the BiLSTM-based contextualized word representations that predated the BERT model (Devlin et al., 2019). Such a comparison had already been done to an extent in English by Peters et al. (2019), but in that case, ELMo and BERT where pre-trained with different datasets, which as we saw in previous chapters, can have an enormous impact on the performance of these types of models. We thus decided to train an ELMo model with the French subcorpus of OSCAR 2019 to fairly compare with CamemBERT. We first compare these two models in a benchmarking experiment in named entity recognition that we do in order to find the best possible combination of embeddings and architectures for NER or at least for the NER annotated version of the FTB that we presented in subsection 9.2.1. We then expand our experiments by actually repeating most of the CamemBERT experiments but this for comparing the OSCAR pre-trained ELMo with CaBeRnet and CBT-fr ELMos.

[1]Contributions: for the part of (Ortiz Suárez et al., 2020a) presented here I pre-trained FrELMo and conducted all the experiments involving FrELMo and CamemBERT. For the part of popa-fabre-etal-2020-french presented here, I pre-trained all the ELMo models and conducted all the evaluations in downstream tasks.

12.1 FrELMo

We train an ELMo model for contemporary French using the French subcorpus of OSCAR 2019. Furthermore, we train each model for 10 epochs, as was done for the original English ELMo (Peters et al., 2018). We also use the same hyper-parameters and the same pre-processing as the originals ELMo authors, i.e., we shuffle the French subcorpus of OSCAR 2019 at a line level. In this case we do not bother to save checkpoints as we previously saw that training for longer produced better models (see 5), so we train for the full 10 epochs as the original authors suggested (Peters et al., 2018).

12.1.1 Benchmarking NER Models

Experiments

For our benchmark of NER models for French, we used SEM (Dupont, 2017) as our strong baseline because, to the best of our knowledge, it was the previous state-of-the-art for named entity recognition on the FTB-NE corpus. Other French NER systems are available, such as the one given by SpaCy. However, it was trained on another corpus called WikiNER, making the results non-comparable. We can also cite the system of (Stern et al., 2012). This system was trained on another newswire (AFP) using the same annotation guidelines, so the results given in this article are not directly comparable. This model was trained on FTB-NE in Stern (2013) (table C.7, page 303), but the article is written in French. The model yielded an F1-score of 0.7564, which makes it a weaker baseline than SEM. We can cite yet another NER system, namely grobid-ner.[2] It was trained on the FTB-NE and yields an F1-score of 0.8739, but two things are to be taken into consideration in grobid-ner's score: the tagset was slightly modified and scores were averaged over a 10-fold cross validation. To see why this is important for FTB-NE, see section 12.1.3.

In this section, we will compare our strong baseline with a series of neural models. We will use the two current state-of-the-art neural architectures for NER, namely seq2seq and LSTM-CRFs models. We will use various pre-trained embeddings in said architectures: fastText, CamemBERT and FrELMo embeddings.

SEM

SEM (Dupont, 2017) is a tool that relies on linear-chain CRFs (Lafferty et al., 2001) to perform tagging. SEM uses Wapiti (Lavergne et al., 2010) v1.5.0 as linear-chain CRFs implementation. SEM uses the following features for NER:

- token, prefix/suffix from 1 to 5 and a Boolean isDigit features in a [-2, 2] window;

[2] https://github.com/kermitt2/grobid-ner#corpus-lemonde-ftb-french

- previous/next common noun in sentence;
- 10 gazetteers (including NE lists and trigger words for NEs) applied with some priority rules in a [-2, 2] window;
- a "fill-in-the-gaps" gazetteers feature where tokens not found in any gazetteer are replaced by their POS, as described in (Raymond and Fayolle, 2010). These features used token unigrams and token bigrams in a [-2, 2] a window.
- tag unigrams and bigrams.

We trained our own SEM model by using SEM features on gold tokenization and optimized L1 and L2 penalties on the development set. The metric used to estimate convergence of the model is the error on the development set ($1 - accuracy$). Our best result on the development set was obtained using the rprop algorithm, a 0.1 L1 penalty and a 0.1 L2 penalty.

SEM also uses an NE mention broadcasting post-processing (mentions found at least once are used as a gazetteer to tag unlabeled mentions), but we did not observe any improvement using this post-processing on the best hyperparameters on the development set.

Neural models

In order to study the relative impact of different word vector representations and different architectures, we trained a number of NER neural models that differ in multiple ways. They use zero to three of the following vector representations: FastText non-contextual embeddings (Bojanowski et al., 2017), the FrELMo contextual language model, and one of multiple CamemBERT language models (Martin et al., 2020) (see Appendix E). The CamemBERT models we use in our experiments differ in multiple ways:

- Training corpus: OSCAR 2019 or CCNet (Wenzek et al., 2020). For comparison purposes, we also display the results of an experiment using the mBERT multilingual BERT model trained on the Wikpiedias for over 100 languages.
- Model size: following Devlin et al. (2019), we use both "BASE" and "LARGE" models; these models differ by their number of layers (12 vs. 24), hidden dimensions (768 vs. 1024), attention heads (12 vs. 16) and, as a result, their number of parameters (110M vs. 340M).
- Masking strategy: the objective function used to train a CamemBERT model is a masked language model objective. However, BERT-like architectures like CamemBERT rely on a fixed vocabulary of explicitly predefined size obtained by an algorithm that splits rarer words into subwords, which are part of the vocabulary together with more frequent words. As a result, it is possible to

use a whole-word masked language objective (the model is trained to guess missing words, which might be made of more than one subword) or a subword masked language objective (the model is trained to guess missing subwords). Our models use the acronyms WWM and SWM respectively to indicate the type of masking they used.

We use these word vector representations in three types of architectures:

- Fine-tuning architectures: in this case, we add a dedicated linear layer to the first subword token of each word, and the whole architecture is then fine-tuned to the NER task on the training data.

- Embedding architectures: word vectors produced by language models are used as word embeddings. We use such embeddings in two types of LSTM-based architectures: an LSTM fed to a seq2seq layer and an LSTM fed to a CRF layer. In such configurations, the use of several word representations at the same time is possible, using concatenation as a combination operator. For instance, in Table 12.1, the model FastText + CamemBERT$_{\text{OSCAR-BASE-WWM}}$ under the header "*LSTM-CRF + embeddings* corresponds to a model using the LSTM-CRF architecture and, as embeddings, the concatenation of FastText embeddings, the output of the CamemBERT "BASE" model trained on OSCAR with a whole-word masking objective, and the output of the FrELMo language model. For all LSTM-based architectures we use the implementation of Straková et al. (2019).

For our neural models, we optimized hyperparameters using F1-score on development set as our convergence metric.

We train each model three times with three different seeds, select the best seed on the development set, and report the results of this seed on the test set in Table 12.1.

12.1.2 Results

Word Embeddings: Results obtained by SEM and by our neural models are shown in table 12.1. First important result that should be noted is that LSTM+CRF and LSTM+seq2seq models have similar performances to that of the SEM (CRF) baseline when they are not augmented with any kind of embeddings. Just adding classical fastText word embeddings dramatically increases the performance of the model.

ELMo Embeddings: Adding contextualized ELMo embeddings increases again the performance for both architectures. However, we note that the difference is not as big as in the case of the pair with/without fastText word embeddings for the LSTM-CRF. For the seq2seq model, it is the contrary: adding ELMo gives a good improvement while fastText does not improve the results as much.

Model	Precision	Recall	F1-Score
			baseline
SEM (CRF)	87.18	80.48	83.70
LSTM-seq2seq	85.10	81.87	83.45
+ FastText	86.98	83.07	84.98
+ FastText + FrELMo	89.49	87.48	88.47
+ FastText + CamemBERT$_{\text{OSCAR-BASE-WWM}}$	89.79	88.86	89.32
+ FastText + CamemBERT$_{\text{OSCAR-BASE-WWM}}$ + FrELMo	90.00	88.60	89.30
+ FastText + CamemBERT$_{\text{CCNET-BASE-WWM}}$	90.31	89.29	89.80
+ FastText + CamemBERT$_{\text{CCNET-BASE-WWM}}$ + FrELMo	90.11	88.86	89.48
+ FastText + CamemBERT$_{\text{OSCAR-BASE-SWM}}$	90.09	89.46	89.77
+ FastText + CamemBERT$_{\text{OSCAR-BASE-SWM}}$ + FrELMo	90.11	88.95	89.53
+ FastText + CamemBERT$_{\text{CCNET-BASE-SWM}}$	90.31	89.38	89.84
+ FastText + CamemBERT$_{\text{CCNET-BASE-SWM}}$ + FrELMo	90.64	89.46	<u>90.05</u>
+ FastText + CamemBERT$_{\text{CCNET-500K-WWM}}$	<u>90.68</u>	89.03	89.85
+ FastText + CamemBERT$_{\text{CCNET-500K-WWM}}$ + FrELMo	90.13	88.34	89.23
+ FastText + CamemBERT$_{\text{CCNET-LARGE-WWM}}$	90.39	88.51	89.44
+ FastText + CamemBERT$_{\text{CCNET-LARGE-WWM}}$ + FrELMo	89.72	88.17	88.94
			LSTM-CRF + embeddings
LSTM-CRF	85.87	81.35	83.55
+ FastText	88.53	84.63	86.53
+ FastText + FrELMo	88.89	88.43	88.66
+ FastText + CamemBERT$_{\text{OSCAR-BASE-WWM}}$	90.47	88.51	89.48
+ FastText + CamemBERT$_{\text{OSCAR-BASE-WWM}}$ + FrELMo	89.70	88.77	89.24
+ FastText + CamemBERT$_{\text{CCNET-BASE-WWM}}$	90.24	89.46	89.85
+ FastText + CamemBERT$_{\text{CCNET-BASE-WWM}}$ + FrELMo	89.38	88.69	89.03
+ FastText + CamemBERT$_{\text{OSCAR-BASE-SWM}}$	**90.96**	<u>89.55</u>	**90.25**
+ FastText + CamemBERT$_{\text{OSCAR-BASE-SWM}}$ + FrELMo	89.44	88.51	88.98
+ FastText + CamemBERT$_{\text{CCNET-BASE-SWM}}$	90.09	88.69	89.38
+ FastText + CamemBERT$_{\text{CCNET-BASE-SWM}}$ + FrELMo	88.18	87.65	87.92
+ FastText + CamemBERT$_{\text{CCNET-500K-WWM}}$	89.46	88.69	89.07
+ FastText + CamemBERT$_{\text{CCNET-500K-WWM}}$ + FrELMo	90.11	88.86	89.48
+ FastText + CamemBERT$_{\text{CCNET-LARGE-WWM}}$	89.19	88.34	88.76
+ FastText + CamemBERT$_{\text{CCNET-LARGE-WWM}}$ + FrELMo	89.03	88.34	88.69
			fine-tuning
mBERT	80.35	84.02	82.14
CamemBERT$_{\text{OSCAR-BASE-WWM}}$	89.36	89.18	89.27
CamemBERT$_{\text{CCNET-500K-WWM}}$	89.35	88.81	89.08
CamemBERT$_{\text{CCNET-LARGE-WWM}}$	88.76	**89.58**	89.39

Table 12.1: Results on the test set for the best development set scores.

CamemBERT Embeddings: Adding the CamemBERT embeddings always increases the performance of the model LSTM based models. However, as opposed to adding ELMo, the difference with/without CamemBERT is equally considerable for both the LSTM-seq2seq and LSTM-CRF. In fact adding CamemBERT embeddings increases the original scores far more than ELMo embeddings does, so much so that the state-of-the-art model is the LSTM + CRF + FastText + $\text{CamemBERT}_{\text{OSCAR-BASE-SWM}}$.

CamemBERT + FrELMo: Contrary to the results given in Straková et al. (2019), adding ELMo to CamemBERT did not have a positive impact on the performances of the models. Our hypothesis for these results is that, contrary to Straková et al. (2019), we trained ELMo and CamemBERT on the same corpus. We think that, in our case, ELMo either does not bring any new information or even interfere with CamemBERT.

Base vs large: an interesting observation is that using large model negatively impacts the performances of the models. One possible reason could be that, because the models are larger, the information is more sparsely distributed and that training on the FTB-NE, a relatively small corpus, is harder.

12.1.3 Impact of shuffling the data

One important thing about the FTB is that the underlying text is made of articles from the newspaper Le Monde that are chronologically ordered. Moreover, the standard development and test sets are at the end of the corpus, which means that they are made of articles that are more recent than those found in the training set. This means that a lot of entities in the development and test sets may be new and therefore unseen in the training set. To estimate the impact of this distribution, we shuffled the data, created a new training/development/test split of the same lengths as in the standard split, and retrained and reevaluated our models. We repeated this process 3 times to avoid unexpected biases. The raw results of this experiment are given in table 12.2. We can see that the shuffled splits result in improvements on all metrics, the improvement in F1-score on the test set ranging from 4.04 to 5.75 (or 25% to 35% error reduction) for our SEM baseline, and from 1.73 to 3.21 (or 18% to 30% error reduction) for our LSTM-CRF architectures, reaching scores comparable to the English state-of-the-art. This highlights a specific difficulty of the FTB-NE corpus where the development and test sets seem to contain non-negligible amounts of unknown entities. This specificity, however, allows to have a quality estimation which is more in line with real use cases, where unknown NEs are frequent. This is especially the case when processing newly produced texts with models trained on FTB-NE, as the text annotated in the FTB is made of articles around 20 years old.

Model	Precision	Recall	F1-Score
			shuf 1
SEM(dev)	92.96	87.84	90.33
LSTM-CRF+CamemBERT$_{\text{OSCAR-BASE-SWM}}$(dev)	93.77	94.00	93.89
SEM(test)	91.88	87.14	89.45
LSTM-CRF+CamemBERT$_{\text{OSCAR-BASE-SWM}}$(test)	**92.59**	**93.96**	**93.27**
			shuf 2
SEM(dev)	91.67	85.96	88.73
LSTM-CRF+CamemBERT$_{\text{OSCAR-BASE-SWM}}$(dev)	93.15	94.21	93.68
SEM(test)	90.57	87.76	89.14
LSTM-CRF+CamemBERT$_{\text{OSCAR-BASE-SWM}}$(test)	**92.63**	**94.31**	**93.46**
			shuf 3
SEM(dev)	92.53	88.75	90.60
LSTM-CRF+CamemBERT$_{\text{OSCAR-BASE-SWM}}$(dev)	94.85	95.82	95.34
SEM(test)	90.68	85.00	87.74
LSTM-CRF+CamemBERT$_{\text{OSCAR-BASE-SWM}}$(test)	**91.30**	**92.67**	**91.98**

Table 12.2: Results on the test set for the best development set scores.

12.1.4 Conclusions of the Benchmark

We establish a new state-of-the-art for French NER using state-of-the-art neural techniques and recently produced neural language models for French. Our best neural model reaches an F1-score which is 6.55 points higher (a 40% error reduction) than the strong baseline provided by the SEM system.

We also highlight how the FTB-NE is a good approximation of a real use case. Its chronological partition increases the number of unseen entities allows to have a better estimation of the generalization capacities of machine learning models than if it were randomized.

One interesting point to investigate is that using Large embeddings overall has a negative impact on the models performances. It could be because larger models store information relevant to NER more sparingly, making it harder for trained models to capitalize them. We would like to investigate this hypothesis in future research.

12.2 Pre-training Corpora Evaluation for ELMo models

Having completed this Benchmark in NER, we also wanted to better understand the computational impact of the quality, size and linguistic balance in ELMo's (Peters et al., 2018) pre-training. We conducted this experiments with ELMo instead of BERT or RoBERTa, as ELMo is a far less demanding model in terms of computing power when it comes to pre-training, and at the moment when we conducted these experiments we didn't have access to the infrastructure required to pre-train Transformer-based models.

12.2.1 ELMo Pre-traing & Fine-tuning Method

Two protocols were carried out to evaluate the impact of corpora characteristics on the tasks under analysis. *Method 1* implies a full pre-training ELMo-based language models for each of the corpora mentioned in Table 9.3. While *Method 2* is based on pre-training OSCAR + fine-tuning with our French Balanced Reference Corpus CaBeRnet, yielding $\text{ELMo}_{\text{OSCAR+CaBeRnet}}$. Hence, the pure pre-traing (i.e. Method 1) yields the following four language models which were pre-trained on the four corpora under comparison : $\text{ELMo}_{\text{OSCAR}}$ (FrELMo in the previous section), $\text{ELMo}_{\text{Wikipedia}}$, $\text{ELMo}_{\text{CaBeRnet}}$ and ELMo_{CBT}.

We conduct the same experiments that we did for CamemBERT in dependency parsing, POS tagging and NER. We also coupled our ELMo models with the same tasks specific architectures as before, namely textbfUDPipe Future (Straka, 2018) for POS tagging and dependency parsing and (Straková et al., 2019) for NER. Experiments were run using the Universal Dependencies (UD) paradigm and its corresponding UD POS-tag set (Petrov et al., 2012) and UD treebank collection version 2.2 (Nivre et al., 2018), which was used for the CoNLL 2018 shared task.

12.2.2 Results & Discussion

Dependency Parsing and POS-tagging

Model	GSD			Sequoia			Spoken			ParTUT		
	UPOS	UAS	LAS	UPOS	UAS	LAS	UPOS	UAS	LAS	UPOS	UAS	LAS
Baseline UDPipe Future	97.63	90.65	88.06	98.79	92.37	90.73	95.91	82.90	77.53	96.93	92.17	89.63
+ELMo_{CBT}	97.49	90.21	87.37	98.40	92.18	90.56	96.60	85.05	79.82	97.27	92.55	90.44
+$\text{ELMo}_{\text{Wikipedia}}$	97.92	92.13	89.77	99.22	94.28	92.97	97.28	85.61	80.79	**97.62**	94.01	91.78
+$\text{ELMo}_{\text{CaBeRnet}}$	97.87	92.02	89.62	99.33	94.42	93.14	***97.30***	85.39	80.63	97.43	94.02	91.86
+$\text{ELMo}_{\text{OSCAR}}$	97.85	92.41	90.05	99.30	94.43	93.25	97.10	85.83	**80.94**	97.47	**94.74**	**92.55**
+$\text{ELMo}_{\text{OSCAR+CaBeRnet}}$	**97.98**	**92.57**	**90.22**	**99.34**	**94.51**	**93.38**	97.24	**85.91**	80.93	97.58	94.47	92.05
State-of-the-art												
UDify	97.83	93.60	91.45	97.89	92.53	90.05	96.23	85.24	80.01	96.12	90.55	88.06
UDPipe Future + mBERT	97.98	92.55	90.31	*99.32*	94.88	93.81	97.23	*86.27*	*81.40*	*97.64*	94.51	92.47
CamemBERT	*98.19*	*94.82*	*92.47*	99.21	*95.56*	*94.39*	96.68	86.05	80.07	97.63	95.21	*92.90*

Table 12.3: Final POS and dependency parsing scores on 4 French treebanks (French GSD, Spoken, Sequoia and ParTUT), reported on test sets (4 averaged runs) assuming gold tokenisation. Best scores in bold, second to best underlined, state-of-the-art results in italics.

$\text{ELMo}_{\text{CaBeRnet}}$: A Test for Balance The representations offered by $\text{ELMo}_{\text{CaBeRnet}}$ are not only competitive but sometimes better than Wikipedia ones. One should keep in mind that almost all the four treebanks we use in this section include Wikipedia data. $\text{ELMo}_{\text{CaBeRnet}}$ is reaching state-of-the-are results in POS-tagging on Spoken. Notably, it performs better than CamemBERT, the previous state of the art on this oral specialized tree-bank (cf. dark gray highlight on Table 12.3). We understand

this results as a clear effect of balance when testing upon a purely spoken test-set. Importantly, this effect is difficultly explainable by the size of oral-style data in CaBeRnet. The oral sub-part is only one fifth of the total, and in this one fifth, only an even smaller amount of data comes from purely oral transcripts comparable the ones in the Spoken tree-bank, namely 67,444 words from Rhapsodie corpus, and 575,894 words form ORFEO. Hence, CaBeRnet's balanced oral language use shows to pay off in POS-tagging. These results are surprising, especially given the fact that our evaluation method was aiming at comparing the quality of word-embedding representations and not beating the state-of-the-art.

$\text{ELMo}_{\text{CaBeRnet}}$: A Test for Coverage From Table 12.3, we discover that not only balance, but also the broad and diverse genre converge of CaBeRnet may play a role in its POS-tagging success is we compare its results with ELMo_{CBT} that also features oral dialogues in youth literature. The fact that ELMo_{CBT} does not show a comparable performance in POS-tagging, can be interpreted as linked to its size, but possibly also to its lack of variety in genres, thus, suggesting the advantage of a comprehensive coverage of language use. This suggests that a balanced sample may enhance the convergence of generalization about oral-style from distinct genre that still implies oral-like dialogues like in fiction. In sum, broad coverage may contribute to enhancing representations about oral language.

The effect of balance on Fine-tuning For POS-tagging in GSD the results of $\text{ELMo}_{\text{OSCAR}}$ are in second place position compared to $\text{ELMo}_{\text{OSCAR+CaBeRnet}}$ that is extremely close to $\text{ELMo}_{\text{Wikipedia}}$. While in POS-tagging in ParTUT, $\text{ELMo}_{\text{Wikipedia}}$ exhibits better results than $\text{ELMo}_{\text{OSCAR}}$, and $\text{ELMo}_{\text{OSCAR+CaBeRnet}}$ is in second position.

Comparing GSD and Sequoia scores from $\text{ELMo}_{\text{OSCAR}}$ and $\text{ELMo}_{\text{OSCAR+CaBeRnet}}$, we observe that fine-tuning with CaBeRnet the embeddings that were pre-trained on OSCAR, yields better representations for the three tasks compared to both the original $\text{ELMo}_{\text{OSCAR}}$ and $\text{ELMo}_{\text{CaBeRnet}}$. However, fine-tuning does not always yield better findings than $\text{ELMo}_{\text{OSCAR}}$ on Spoken and ParTUT, where $\text{ELMo}_{\text{OSCAR+CaBeRnet}}$ places in second after $\text{ELMo}_{\text{OSCAR}}$ for parsing scores UAS/LAS (cf. Table 12.3).

A closer look on Parsing results reveals an interesting pattern of results across treebanks (see light gray highlights on Table 12.3). We see that for GSD and Sequoia the CaBeRnet fine-tuned version $\text{ELMo}_{\text{OSCAR+CaBeRnet}}$ compared to the pure OSCAR pre-trained $\text{ELMo}_{\text{OSCAR}}$ is achieving higher scores. While a reverse and less clear-cut pattern is observable for the other two treebanks, namely Spoken and ParTUT. This configuration can be explained if we understand this pattern as due to the reinforcement and unlearning of $\text{ELMo}_{\text{OSCAR}}$ representations during the process of fine-tuning. Specifically, we can observe that parsing scores are better on treebanks that share the kind of language use represented in CaBeRnet, while they are worse

on corpora that are closer in language sample to OSCAR corpus, like Spoken and ParTuT. This calls for further developments of CaBeRnet (§12.2.3).

ELMo$_{CBT}$: SMALL BUT RELEVANT ELMo$_{CBT}$ shows an intriguing pattern of results. Even if its scores are under the baseline on GSD and Sequoia, it yields over the baseline results for Spoken and ParTUT. Given its reduced size, one would expect it to overfit, this would explain the under baseline performance. However, this was not the case on Spoken and ParTUT treebanks, thus showing ELMo$_{CBT}$ contribution in generating representations that are useful to UDPipe model to achieve better results in POS-tagging and parsing tasks on the ParTUT and Spoken tree-banks. The presence of oral dialogues is certainly playing a role in this results' pattern. This unexpected result calls for further investigation on the impact of pre-training with reduced-size, noiseless, domain-specific corpora.

NER

NER - RESULTS on FTB	Precision	Recall	F1
Baselines Models			
SEM (CRF) (Dupont, 2017)	87.89	82.34	85.02
LSTM-CRF (Dupont, 2017)	87.23	83.96	85.57
LSTM-CRF test models	85.87	81.35	83.55
+FastText	88.53	84.63	86.53
+FastText+ELMo$_{CBT}$	79.77	77.63	78.69
+FastText+ELMo$_{Wikipedia}$	88.87	87.56	88.21
+FastText+ELMo$_{CaBeRnet}$	88.91	87.22	88.06
+FastText+ELMo$_{OSCAR}$	88.89	88.43	88.66
+FastText+ELMo$_{OSCAR+CaBeRnet}$	**90.70**	**89.12**	**89.93**
State-of-the-art Models			
CamemBERT (Martin et al., 2020)	89.35	88.81	89.08

Table 12.4: NER Results on French Treebank (FTB): **best scores**, second to best.

For NER, LSTM-CRF+FastText+ELMo$_{OSCAR+CaBeRnet}$ achieves a better precision, recall and F1 than the traditional CRF-based SEM architectures and even CamemBERT. Importantly, LSTM-CRF+FastText+ELMo$_{CaBeRnet}$ reaches better results in finding entity mentions, than Wikipedia which is a highly specialized corpus in terms of vocabulary variety and size, as can be seen in the overwhelming total number of unique forms it contains (see Table 9.4). We can conclude that both pre-training and fine-tuning with CaBeRnet on ELMo$_{OSCAR}$ generates better word-embedding representations than Wikipedia in this downstream task.

CBT-fr NER results are under the LSTM-CRF baseline. This can possibly be explained by the distance in terms of topics and domain from FTB treebank (i.e.

newspaper articles), or by the reduced-size of the corpus to yield good-enough representation to perform named entity recognition.

All in all, our evaluations confirm the effectiveness of large ELMo-based language models fine-tuned or pre-trained with a balanced and linguistically representative corpus, like CaBeRnet as opposed to domain-specific ones.

12.2.3 Conclusion

We investigated the relevance of different types of corpora on ELMo's pre-training and fine-tuning. It confirms the effectiveness and quality of word-embeddings obtained through balanced and linguistically representative corpora.

The proposed evaluation methods are showing that CaBeRnet and CBT-fr are not only relevant for neural NLP and language modeling in French, but that corpus balance shows to be a significant predictor of ELMo's accuracy on Spoken test data-set and for NER tasks.

The results obtained for the parsing tasks on ParTUT open a new perspective for the development of the French Balanced Reference Corpus, involving the enhancement of the terminological coverage of CaBeRnet. A sixth sub-part could be included to cover technical domains like legal and medical ones, and thereby enlarge the specialized lexical coverage of CaBeRnet.

Further developments of this resource would involve an extension to cover user-generated content, ranging from well written blogs, tweets to more variable written productions like newspaper's comment or forums, as present in the CoMeRe corpus (Chanier et al., 2014). The computational experiments conducted here also show that pre-training language models like ELMo on a very small sample like the French Children Book Test corpus or CaBeRnet yields unexpected results. This opens a perspective for languages that have smaller training corpora. ELMo could be a better suited language model for those languages than it is for others having larger size resources.

To conclude, our current evaluations show that linguistic quality in terms of *representativeness* and balance yields better performing contextualized word-embeddings.

13 SinNer CLEF-HIPE2020

In which we present our participation (Ortiz Suárez et al., 2020) to the CLEF-HIPE 2020 shared task (Ehrmann et al., 2020). We actually use this NER shared task as an opportunity see how well the Contemporary French language models that we had trained could perform without any further pre-training or fine-tuning in a task that mainly used historical data. Here we also evaluate the impact that pre-processing has on the models' performance.[1]

Having participated in the pre-training of Contemporary French language models like CamemBERT (Martin et al., 2020) and FrELMo (Ortiz Suárez et al., 2020a; Popa-Fabre et al., 2020) we wanted to better asses how these models would perform in downstream tasks without any further fine-tuning or additional pre-training data. We wanted to assess in particular how much effective would be a transfer learning technique between different states of the French language. The CLEF-HIPE 2020 shared task (Ehrmann et al., 2020) seemed then like a good opportunity to test this, as HIPE (Identifying Historical People, Places and other Entities) was an evaluation campaign on named entity processing on historical newspapers in French, German and English, which was organized in the context of the impresso project and run as a CLEF 2020 Evaluation Lab. In this study we also evaluate the impact that pre-processing has on the performance of our models, specially when the training data comes from OCRized text.

13.1 Dataset for the CLEF-HIPE shared task

The dataset of the CLEF-HIPE shared task contains newspaper articles of 17th-20th century. The text is an output of an OCR software, then tokenized and annotated with labels corresponding to each sub-task. This peculiarity of historical documents will be detailed later in this section. The corpus provided for French and German both contained training data (train) and development data (dev) whereas, for English only development data was provided for the shared task. For this reason, we chose to work on French and German only.

[1]Contributions: I made all the experiments for French and German involving the ELMo models and the LSTM-CRFs. Notably the runs 1 and 2. I also actively contributed to writing the scientific article.

Table 13.1 shows some statistics of this dataset. The size of the training dataset for French was twice as big as for German, whereas the development sets had roughly the same size. As usual in NER, persons (Pers) and locations (Loc) were the most frequent entity types.

	Tokens	Documents	Segments	Labeled named entities				
				Pers	Loc	Org	Time	Prod
Train Fr	166217	158	19183	3067	2513	833	273	198
Dev Fr	37592	43	4423	771	677	158	69	48
Train De	86960	104	10353	1747	1170	358	118	112
Dev De	36175	40	4186	664	428	172	73	53

Table 13.1: Statistics on the training and development data in French and German

TOKEN	NE-COARSE		NE-FINE			NE-NESTED	NEL		MISC
	LIT	METO	LIT	METO	COMP		LIT	METO	
# language = fr									
# newspaper = EXP									
# date = 1918-04-22									
# document_id = EXP-1918-04-22-a-i0077									
# segment_iiif_link = https://iiif.dhlab.epfl.ch/iiif_impresso...									
Lettre	O	O	O	O	O	O	_	_	_
de	O	O	O	O	O	O	_	_	_
la	O	O	O	O	O	O	_	_	_
Su	B-loc	O	B-loc.adm.reg	O	O	B-loc.adm.nat	Q689055	_	NoSpaceAfter
.	I-loc	O	I-loc.adm.reg	O	O	I-loc.adm.nat	Q689055	_	_
_	I-loc	O	I-loc.adm.reg	O	O	I-loc.adm.nat	Q689055	_	NoSpaceAfter
sss	I-loc	O	I-loc.adm.reg	O	O	I-loc.adm.nat	Q689055	_	_
allemands	I-loc	O	I-loc.adm.reg	O	O	O	Q689055	_	EndOfLine
# segment_iiif_link = https://iiif.dhlab.epfl.ch/iiif_impresso...									
(	O	O	O	O	O	O	_	_	NoSpaceAfter
Nous	O	O	O	O	O	O	_	_	_
serons	O	O	O	O	O	O	_	_	_
heureux	O	O	O	O	O	O	_	_	_
de	O	O	O	O	O	O	_	_	_
publier	O	O	O	O	O	O	_	_	_
...									

Table 13.2: Example extracted from the French training dataset

Table 13.2 shows an excerpt of the train dataset (CoNLL format). For each document, general information were provided. Among them, newspaper and date may have been features useful for recognizing entities, but we did not take advantage of it. Each document was composed of segments, starting with "# segment ..." corresponding to lines in the original documents. Each segment is tokenized in order to correspond to the CoNLL format with one token per line. These two notions, segments and tokens, are very important since they do not always match the type of unit usually processed in NLP pipelines. Segments seldom correspond to sentences, so there is a need to concatenate the segments to get the raw text and then segment it into sentences. This is very interesting since it gets us close to real-world conditions rather than laboratory conditions, and as we show in Section 13.3.2, that this segment vs. sentence question has an important influence on the results. Regarding tokens,

the tokenization is obviously not perfect. We can see that there are non-standard words and bad tokenization due to the OCR output (in red in Table 13.2). If we concatenate the tokens we get the sequence "Su. _sss allemands" instead of "Suisse allemande". These non-standard words make the Named Entity Recognition task more complicated and, again, more realistic.

13.2 CRFs and Contextualized Word Embeddings for NER

Here we present the different models that we use for the shared-tasks, which are again some of the architectures used in the previous chapter for the NER benchmark.

13.2.1 CRF model (run3)

We use SEM (Segmenteur-Étiqueteur Markovien)[2,3] (Dupont, 2017) with the exact same features described in Subsection 12.1.1.

We trained a CLEF HIPE specific model by optimizing L1 and L2 penalties on the development set. The metric used to estimate convergence of the model is the error on the development set ($1 - accuracy$). For French, our optimal L1 and L2 penalties were 0.5 and 0.0001 respectively (default Wapiti parameters). For German, our optimal L1 and L2 penalties were 1.0 and 0.0001 respectively.

One interest of SEM is that it has a built-in sentence tokenizer for French using a rule-based approach. By default, CLEF-HIPE provides a newline segmentation that is the output of an OCR engine. As a result, some NE mentions span across multiple segments, making it very hard to identify them correctly. It is to be expected that models trained (and labelling on) sentences would yield better performances than those trained (and labelling on) segments. SEM makes it simple to switch between different sequence segmentations, which allowed us to label sentences and output segments. SEM's sentence segmentation engine works using mainly local rules to determine whether a token is the last of a sequence (e.g.: is a dot preceded by a known title abbreviation?). It also uses non-local rules to remember whether a token is between parentheses or French quotes to not segment automatically within them. Since we work at token level, we had to adapt some rules to fit CLEF-HIPE tokenization. For example, SEM decides at tokenization stage whether a dot is a strong punctuation or part of a larger token, as for abbreviations. This has the advantage of making sentence segmentation easier. CLEF-HIPE tokenization systematically separates dots, so we adapted some sentence segmentation rules, for example: we decided not to consider a dot as a sentence terminator if the previous token was in a lexicon of titles or functions. No specific handling of OCR errors were done. Another interest is that SEM has an NE mention broadcasting process. Mentions found at

[2]available at: https://github.com/YoannDupont/SEM.
[3]translates to: Markovian Tokenizer-Tagger (MTT).

least once in a document are used as a gazetteer to tag unlabeled mentions within said document. When a new mention overlaps and is strictly longer than an already found mention, the new mention will replace the previous one in the document.

13.2.2 ELMo-LSTM-CRF (RUN1 AND RUN2)

For our experiments, we follow the same approach of previous chapters by using the Bi-LSTM-CRF implementation of Straková et al. (2019) which is open source and readily available,[4] and pre-appending contextualized word-embeddings to the model. For French we pre-append the FrELMo model Ortiz Suárez et al. (2020a), which as we saw in the previous chapter, is the standard ELMo (Peters et al., 2018) implementation[5] trained on the French subcorpus of OSCAR 2019[6] (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020b). For German we pre-append the German ELMo (May, 2019), which is again the standard ELMo implementation but trained on the German Wikipedia.

Contrary to the approach used in Subsection 12.1.1, we do not use the CamemBERT model (Martin et al., 2020) for French or the German BERT (Chan et al., 2019). Both of these models are BERT-based and as such they are limited to a 512-token contextualized window. Moreover, they both use SentencePiece (Kudo and Richardson, 2018) meaning that tokens are actually subwords, which considerably increases the number of tokens per sentence, specially for the longer ones, thus decreasing the contextual windows of both CamemBERT and the German BERT. SentencePiece also introduces the problem of a fixed-size vocabulary, which in the case of this shared task might negatively impact the performance of said models, as they could struggle to handle OCR problems or just non-standard vocabulary. Since our main goal was to reconstruct the sentences and use long contextualized sequences we opted to use ELMo which can easily handle longer sequences with its standard implementation and actually has a dynamic vocabulary thanks to the CNN character embedding layer, thus it might be better equipped to handle non-standard orthography and OCR problems. We actually did a preliminary experiment for French with CamemBERT, but the results were even worse than our baselines, so we decided to scrap it. We will see this phenomenon again on chapter 15 where we will actually report it and study it more in depth.

For the fixed word embeddings we used the Common Crawl-based FastText embeddings (Grave et al., 2018) originally trained by Facebook as opposed to the embeddings provided by the HIPE shared task, as we obtained better dev scores using the original FastText embeddings for both French and German.

[4]Available at: https://github.com/ufal/acl2019_nested_ner.
[5]Available at: https://github.com/allenai/bilm-tf.
[6]Available at: https://oscar-corpus.com.

run	French			German		
	P	R	F1	P	R	F1
winner	83.1	84.9	84.0	79.0	80.5	79.7
run 1	77.8	79.4	78.6	63.1	**66.6**	64.8
run 2	**78.8**	**80.2**	**79.5**	**65.8**	65.8	**65.8**
run 3	70.2	57.9	63.5	64.4	43.8	52.1
average	70.2	66.7	67.6	63.8	58.1	60.0
median	71.5	68.6	68.6	66.8	57.7	64.5

Table 13.3: Strict results for our systems compared to the winning system (micro measures)

We used the standard hyperparameters originally[7] used by Straková et al. (2019). Namely a batch size of 8, a dropout of 0.5, a learning rate of 0.001 and 10 epochs. The difference between run 1 and 2, is that run 1 uses the data as is, while run 2 uses the reconstructed sentences.

13.3 Results and Discussion

13.3.1 Official shared task results

The results of our 3 runs compared to the best run on the NERC-coarse shared-task for French and German are given in Table 13.3 (strict scenario). For both tasks, we are the third best ranking team. We only did very minimal adaptation of existing systems. We did not modify tokenization for any language. The most notable change was to use custom sentence segmentation instead of given segments for French and using some additional lexica as features for our CRF model in German (for French, we only used existing SEM lexica). Other than that, we only optimized hyperparameters on the dev set. This clearly illustrates the power of contextual embeddings and today's neural network architectures. This is encouraging in terms of usability of state-of-the-art models on real-world data.

13.3.2 Study of sequence segmentation

In this section, we evaluate the influence of sequence segmentation on system performances. This evaluation is done for French only, as we used SEM to provide sentence segmentation and SEM could only provide a proper sentence segmentation for that language. As can be seen in table 13.4, sentence segmentation allows us to improve results by 3.5 F1 points. This is due to the fact that some entities were split across multiple segments in the original data. Using a custom sentence segmentation allows to have entities in a single sequence. This segmentation is applied both with training

[7] https://github.com/ufal/acl2019_nested_ner/blob/master/tagger.py#L484.

Type	P		R		F1	
	Segments	Sentences	Segments	Sentences	Segments	Sentences
Loc	85.21	87.73 (+2.52)	87.52	87.08 (-0.44)	86.35	87.41 (+1.06)
Org	70.62	71.33 (+0.71)	62.78	65.64 (+2.86)	66.47	68.37 (+1.90)
Pers	80.24	84.64 (+4.40)	76.88	82.09 (+5.21)	78.52	83.35 (+4.83)
Prod	62.96	75.86 (+12.90)	39.53	56.41 (+16.88)	48.57	64.71 (+16.14)
Time	86.21	90.91 (+4.70)	78.12	87.72 (+9.60)	81.97	89.29 (+7.32)
Global	81.03	84.46 (+3.43)	81.61	84.46 (+2.85)	79.52	83.01 (+3.49)

Table 13.4: Comparison between segments and sentences on French dev dataset (run 1), strict scenario

Metric	French		German	
	not to dev	to dev	not to dev	to dev
P	78.8	**79.5** (+0.7)	65.8	**68.2** (+2.4)
R	80.2	**80.7** (+0.5)	65.8	**66.1** (+0.3)
F1	79.5	**80.1** (+0.6)	65.8	**67.1** (+1.3)

Table 13.5: Results obtained on the test set (strict metric) with only the train set (not to dev) and with train+dev sets (to dev) with our best system (run 2)

data and evaluation data, so that our systems can access a more proper context for named entities. The cost of using another segmentation is relatively cheap, as SEM can process nearly 1 GB of raw text per hour.

Per entity comparison is also available in Table 13.4. One can see that the improvement of sentence segmentation is not very significant for locations (Loc). This is due to two facts : (i) locations are usually small in terms of tokens and therefore less prone to be separated in two segments and (ii) there was less room from improvement since they were the easiest entity type to detect (86.35% F1-score). In contrast, entities of type "product" (Prod), usually longer in tokens, were very hard to predict with only 48.57% F1-measure and benefited the most from segmentation in sentences (+16 percentage points in F1-measure).

13.3.3 To dev or not to dev?

In Table 13.5 we show the results that could have been obtained by training the Bi-LSTM model on both train and dev dataset. We used the same hyperparameters as we did for our official run. Despite the fact that it does not ensure the robustness of the system, the added-value seem to be quite disappointing.[8] In German the gain may be a bit more significant, probably due to the smaller size of the training dataset.

[8] In particular, if we consider that it would not have given us a better ranking on any language.

13.4 Conclusion

In this chapter we presented three methods developed for the Named Entity Recognition task in French and German historical newspapers. The first method relied on linear-chain CRFs while the other two methods use a Bidirectional LSTM and a bidirectional Language Model (ELMo). The latter outperformed the CRF model and achieved rank 3 on the NER task in both French and German. We also showed that the type of sequences used has a significant influence on the results. When we segment in sentences rather than using the segments of the dataset as it is the results are systematically much better, with an exception for locations where the gain is marginal. This suggests that sentence segmentation remains a key component of efficient NLP architectures, in particular for models taking advantage of the context.

14 BERTrade

In which we present part of the work of Grobol et al. (2022) who pre-train and develop RoBERTa-based (Liu et al., 2019) models for Medieval French and subsequently evaluate them on POS tagging and dependency parsing on the SRCMF treebank (Prévost and Stein, 2013). Most notably, Grobol et al. (2022) try to transfer knowledge from contemporary French to Old French by post-training existing Contemporary French language models with a small Medieval French corpus.[1]

Having successfully pre-trained and evaluated state-of-the-art language models for Contemporary French, we wanted to finally develop these kinds of resources for Historical French. To this end, we believed Old French to be a particularly interesting language for this kind of study, since relatively to its limited amount of available raw text, its volume of *annotated* linguistic data is quite high, due to the existence of the SRCMF dependency treebank (Prévost and Stein, 2013) and its latest incarnation in the Universal Dependency project (Nivre et al., 2020), which boasts around 17.7 K sentences[2] for around 171 K words.

Another interesting property of Old French is its proximity to a well-resourced language, namely contemporary French, for which we had previously developed monolingual contextual embeddings models (and for which other language models had also been developed at this point) that had been shown to be relevant for dependency parsing as discussed in Chapters 5, 11,12 and on (Martin et al., 2020; Le et al., 2020a). Last, but certainly not least, the design of an accurate syntactic parser for Old French would be a very valuable tool for computer-assisted linguistic studies. Indeed, studying the historical variation of syntax in a language that lacks both native speakers and centralized standard variants can be very challenging, due to the prohibitive cost of manual annotation. Automatic syntactic annotations, either as a "silver-standard" truth or as a bootstrapping step towards manual annotation, can drastically reduce that cost.

In this chapter, exploiting this currently unique situation of Old French among lesser-resourced and historical languages, we use dependency parsing and POS-

[1] Contributions: I participated to the pre-training of some models and their evaluations in downstream tasks, some of which are sadly not presented here as we obtained rather negative results with them. I also did all the computations for the carbon footprint that are presented in appendix F.3.

[2] Putting it in the second place of all French language treebanks in number of sentences.

tagging of Old French as probes of the relevance of contextual embeddings in a context of high heterogeneity and relative scarcity of data. More precisely, we consider several neural language models, some of which we train or fine-tuned on the corpus of raw Old and Middle French texts presented in Section 10.1, and use their internal representations of words as inputs to train taggers and parsers on the SRCMF treebank. The resulting tagging and parsing scores then serve as an evaluation of the quality and usefulness of these representations. In particular:

- We provide empirical evidence that contextual embeddings are relevant for historical language processing, even when no data is available beyond the treebank used to train a parser.
- We provide a comparative study of several strategies for obtaining such contextual embeddings. Specifically, we compare cases where raw data is available in the target language and cases where existing contextual embeddings are available for the contemporary counterpart of a historical language.
- We develop BERTrade,[3] a set of contextual word embedding models and a state-of-the-art POS-tagging and dependency parsing model.

14.1 Experiments

We evaluate a set of alternative word representations on Old French, using their usefulness for POS-tagging and dependency parsing as a downstream evaluation. To that end, we use the annotated treebank of Old French (SRCMF) (Prévost and Stein, 2013) as provided by the 2.7 version of the UD dataset (Zeman et al., 2020) as a reference treebank. We note however that a more recent version of the SRCMF has been recently published (Zeman et al., 2021) after we did these experiments, this new version introduces most notably the punctuation to the SRCMF treebank.

Our parser/tagger probe uses Dozat and Manning (2018)'s neural graph parser made as reimplemented by Le et al. (2020a) and Grobol and Crabbé (2021), using the same hyperparameters. Word representations are obtained by concatenating subword embeddings, averaged over transformer layers together with character embeddings and non contextualized word embeddings. This representation is similar to those used by Straka et al. (2019); Ling et al. (2015). In all of our experiments, the contextual embeddings are fine-tuned while training the parser. Unlike the recent CoNLL challenges settings, we assume gold tokenization, since the syntactic annotations we target provide a reference word-based segmentation. Using a predicted one could only add noise to our experiments. Furthermore, for most European languages using a Latin script—including Old and Middle French—, word segmentation is acceptably approximated by simple typographic tokenization.

[3] *Bertrade de Laon*, also known as *Berthe au Grand Pied* was the mother of Charlemagne.

The remaining of this section presents our experimental results, sorted by nature of required data. We report UPOS POS-tagging scores as well as unlabeled and labeled attachment scores for dependency parsing (respectively UAS and LAS), as given by the CoNLL-2018 scorer, computed on the development set of SRCMF to avoid overfitting the architecture and transfer learning procedure to the test set. Results on the test set are provided only for the dev-best models to allow us to compare our results to the state of the art.

Due to the number of costly experiments,[4] the results are reported on single runs. The results should therefore be interpreted only with respects to the broad trends: small score differences between competing settings should be taken with care.

14.1.1 Baselines

Embeddings	UPOS	UAS	LAS
Vanilla	93.51	87.60	81.54
Random-base	93.17	86.97	80.71
finBERT	**94.44**	**88.44**	**82.47**

Table 14.1: Results on SRCMF dev — no additional data.

We first compare a baseline where contextual embeddings are not used at all (Vanilla) with two settings using models with no preexisting knowledge of Old French: Random-base, a randomly initialized model using the same architecture and model size as RoBERTa-base (Liu et al., 2019) and finBERT (Virtanen et al., 2019), a contextual embedding model from Finnish, a Uralic language that is unrelated to Old French. These baselines are meant to check that the gain in performances observed when using models with some (possibly indirect) knowledge of Old French are linked to this knowledge and not simply due to an increase in the number of trainable parameters (for the random baseline) or to a weight distribution induced by training on a language modeling task that would be universally good for all languages (for the finBERT baseline, which can thus be seen as a different kind of weight initialization).

Table 14.1 shows the results obtained in these configurations, which show that using a model with random weights, even fine-tuned for these tasks, does not bring any improvement, and is in fact even worse than using no contextual embeddings at all. In contrast, using a model that has been pre-trained for language modeling—even for an unrelated language—brings some modest improvements. This suggests that pre-training gives a structure to this kind of model that makes it suitable for fine-tuning on the downstream task, but the impact of this gain is clearly—and

[4] See the Appendix F.3 for elements on the carbon footprint of our experiments.

predictably—very limited compared to what can be expected for representations that have been trained on relevant linguistic data.

14.1.2 With related contextual embeddings

Base model	UPOS	UAS	LAS
FlauBERT	95.70	90.43	85.45
CamemBERT	95.86	91.15	86.31
mBERT	**96.06**	**91.52**	**86.83**

Table 14.2: Results on SRCMF dev — monolingual models.

When a low-resource language is close to a well-resourced one, it is possible to leverage models designed for the latter. For Old French, contemporary French is an obvious candidate and two contextual embeddings models are available: FlauBERT (Le et al., 2020a) and CamemBERT (Martin et al., 2020). Furthermore, mBERT (Devlin et al., 2019), a model trained on a multilingual corpus which does not include Old French (possibly apart from some fragments in its contemporary French training data), has been shown to be suitable for many languages, and in particular for Indo-European and Romance languages (Straka et al., 2019; Muller et al., 2021). We report in table 14.2 the results obtained when using these language models directly, without additional fine-tuning involving Old French data.

As expected, these results show significant improvements over the baselines, confirming that using contextual embeddings for a related language works better than both randomly initialized embeddings and embeddings pre-trained for an unrelated language—even after fine-tuning. More surprisingly, the best results here are obtained with mBERT. This could mean that mBERT benefits from having been pre-trained for a wider range of languages, including in particular other Romance languages that share with Old French some features, for instance null subjects.

14.1.3 With raw linguistic data

We now try to take advantage of the raw Medieval French data described in section 10.1. To that end, we explore two strategies: training a model from scratch and refining existing models by "post-training" them—running a few more training epochs on the Medieval French raw data.

In the "from scratch" strategy we first train a BPE sub-word tokenizer (Wang et al., 2020a) on our raw corpus, then train a RoBERTa (Liu et al., 2019) masked language model. We take inspiration from Micheli et al. (2020), who worked in a setting close to ours: a small and noisy pre-training corpus used to create a model from scratch, we used a RoBERTa architecture. As reported in table 14.3, we tested

Name	Layers	Embeddings	Heads	UPOS	UAS	LAS
BERTrade-tiny	2	128	2	94.03	88.66	82.79
BERTrade-small	4	512	8	96.53	86.30	87.49
BERTrade-petit	12	256	4	**97.14**	**91.90**	**89.18**
BERTrade-medium	8	512	8	96.62	91.92	87.60
BERTrade-base	12	768	12	96.74	92.37	88.42

Table 14.3: Results on SRCMF dev — Performances of different model sizes when training from scratch

several parametrizations of the architecture also inspired by Turc et al. (2019). Out of these alternatives, the "BERTrade-petit" configuration was the most successful and this is the one we keep for the following experiments.

For the "post-training" strategy, we continue the training of the pre-trained models used in sections 14.1.1 and 14.1.2, for 12 epochs on our raw corpus. We used the same RoBERTa masked language modeling task, using the same parameters as Wang et al. (2020b) (but without vocabulary modifications), resulting in the BERTrade-X models, where X is the name of the base model.

Base model	UPOS	UAS	LAS
BERTrade-petit	97.14	92.95	89.18
BERTrade-finBERT	96.28	92.12	87.92
BERTrade-mBERT	96.95	93.33	89.60
BERTrade-CamemBERT	**97.16**	**93.75**	90.06
BERTrade-FlauBERT	96.94	**93.75**	**90.07**

Table 14.4: Results on SRCMF dev — using raw data.

The results of these experiments are reported in Table 14.4. Comparing these to our results of section 14.1.2 shows that training a model from scratch, even on such limited amounts of data, yields a better model than a simple task-specific fine-tuning of mBERT. However, post-training mBERT yields even better results, and the best ones are obtained by post-training the models for contemporary French.

14.1.4 Putting it all together

Finally, in table 14.5, we compare the performances of our models on the test set of SRCMF with those obtained by Straka et al. (2019), with similar methods. The difference between the models is that we fine-tune the word embeddings, while Straka et al. (2019) keep them frozen.

Our mBERT baseline, which is the closest to their configuration, shows that even without any additional data, task-specific fine-tuning already brings significant

Model	UPOS	UAS	LAS
Straka et al. (2019)	96.26	91.83	86.75
mBERT	96.19	92.03	87.52
BERTrade-petit	96.60	92.20	87.95
BERTrade-mBERT	97.11	93.86	90.37
BERTrade-FlauBERT	97.15	93.96	90.57
BERTrade-CamemBERT	**97.29**	**94.36**	**90.90**

Table 14.5: Results on SRCMF test

improvements, while our models refined using our raw corpus of Medieval French bring further improvements, leading to state-of-the-art results that are consistent with their results on the development set.

14.2 Conclusion

In this chapter, we have shown that building a monolingual contextual word embeddings model for Medieval French is possible even with limited and heterogeneous linguistic data and that it can bring significant performance gains in parsing and POS-tagging. To that end, the best strategy seems to be post-training a contextual word embedding model for contemporary French on raw Medieval French documents. We have not directly addressed the internal heterogeneity issue in both our pre-training and fine-tuning data, relying instead on the versatility of the representation models that we considered can bypass it, but it seems a promising perspective for future work—for instance by using finer-grained post-training, concentrating on specific linguistic sub-periods or genres.

For historical languages in general, where not a lot of data is available, this suggests that language-specific fine-tuning is more efficient when applied to a model pre-trained for their contemporary counterpart than when applied to a multilingual model. While this study is not currently easy to replicate for other languages due to the lack of annotated data for a suitable downstream task, it suggests that the considerable amount of work required to gather even a small amount of raw texts in the target language is a sound investment, given the significant improvements it can bring to contextual word representations. Beyond historical languages, these findings could also help for processing minority dialectal variants and contact languages of well-resourced languages, and we leave for future work the exploration of these generalizations.

15 D'AlemBERT

In which we present part of the work of Gabay et al. (2022) who pre-train and develop RoBERTa-based (Liu et al., 2019) models for Early Modern French from scratch with the FreEM$_{max}$ corpus, and subsequently evaluate it on POS tagging and named entity recognition on the FreEM$_{LPM}$ and the FreEM$_{NER}$ corpora respectively.[1]

After having successfully pre-trained and evaluated a Transformer-based language model for Medieval French, we wanted to develop such a model for Early Modern French, the state of language corresponding to that of the *Dictionnaire Universel* in its 1701 edition (Furetière, 1701), the main text of study of the ANR BASNUM (ANR-18-CE38-0003) that funded this Ph.D. thesis. Thus, in this chapter we develop D'AlemBERT, a neural language model for Early Modern French, and we evaluate it in POS tagging and NER on the FreEM$_{LPM}$ and the FreEM$_{NER}$ corpora respectively. Contrary to the approach used in the previous chapter, we only pre-train D'AlemBERT from scratch, and we do not post-train any of the Contemporary French models. We decided to do this mainly because that our FreEM$_{max}$, our pre-training dataset is around 1.2 GB in size which is around 20 times the size of the corpus used in the BERTrade experiments for Medieval French, we also thought that 1.2 GB would be enough to properly train a RoBERTa-based architecture in light of the results obtained for CamemBERT in subsection 11.3.2.

15.1 D'AlemBERT: a neural language model for Early Modern French

In this section, we describe the pre-training data, architecture, training objective and optimization setup we use for D'AlemBERT, our new neural language model for Early Modern French.

[1] Contributions: I pre-trained D'AlemBERT and did all the experiments in downstream tasks as well as the fine-tuning of CamemBERT. I also actively participated in writing the scientific article.

15.1.1 Pre-processing

Similar to RoBERTa (Liu et al., 2019) we segment the input text data into subword units using Byte-Pair encoding (BPE) (Sennrich et al., 2016) in the implementation proposed by (Radford et al., 2019) that uses bytes instead of unicode characters as the base subword units. The BPE encoding does not require pre-tokenization (at the word or token level), thus removing the need to develop a specific tokenizer for Early Modern French. We use a vocabulary size of 32,768 subword tokens. These subwords are learned on the entire FreEM$_{max}$ dataset.

15.1.2 Language Modelling

Transformer D'AlemBERT uses the exact same architecture as RoBERTa, which is a multi-layer bidirectional Transformer (Vaswani et al., 2017). D'AlemBERT uses the original *base* architecture of RoBERTa (12 layers, 768 hidden dimensions, 12 attention heads, 110M parameters).

Pre-training Objective We train our model on the Masked Language Modelling (MLM) task as proposed by RoBERTa's authors (Liu et al., 2019): given an input text sequence composed of N tokens $x_1, ..., x_N$, we select 15% of tokens for possible replacement. Among those selected tokens, 80% are replaced with the special `<MASK>` token, 10% are left unchanged and 10% are replaced by a random token. The model is then trained to predict the masked tokens using cross-entropy loss.

Again, following the RoBERTa approach, we dynamically mask tokens instead of fixing them statically for the whole dataset during preprocessing. We also choose not to use the next sentence prediction (NSP) task originally used in BERT (Devlin et al., 2019), as it has been shown that it does not improve downstream task performance (Conneau and Lample, 2019; Liu et al., 2019).

Optimization Optimization for our model in the exact same way as (Liu et al., 2019) using Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.98$) for 31k steps with large batch sizes of 8,192 sequences, each sequence containing at most 512 tokens.

Pre-training We use the RoBERTa implementation in the Zelda Rose library,[2] and again, in the same way as Liu et al. (2019) our learning rate is warmed up for 10k steps up to a peak value of 0.0003 instead of the original 0.0001 used by the original implementation of RoBERTa (Liu et al., 2019), as our model diverged with the 0.0001 value. Furthermore, we hypothesize that this is either due to the smaller size of FreEM$_{max}$ (compared to the corpora used for RoBERTa or CamemBERT) or to our large batch size. We train our model for 31k steps, which amounts to 41 epochs. The

[2] https://github.com/LoicGrobol/zeldarose

total pre-training times, the details of the infrastructure we used and even the carbon emissions of our model are reported in Appendix G.1.

15.2 Evaluation and Discussion

15.2.1 Part-Of-Speech Tagging

In order to evaluate our D'AlemBERT model, we first fine-tune it for POS tagging on the FreEM$_{LPM}$ corpus. We use the `flair` framework[3] for sequence tagging (Akbik et al., 2019). To fine-tune D'AlemBERT for POS we follow the same approach as Schweter and Akbik (2020) with some modifications: we append a linear layer of size 256 that takes as input the last hidden representation of the `<s>` special token and the mean of the last hidden representation of the subword units of each token (token as defined for FreEM$_{LPM}$), that is, we use a *"mean"* subword pooling strategy. We fine-tune D'AlemBERT with a learning rate of 0.000005 for a total of 10 epochs. We also fine-tune CamemBERT using the exact same hyperparameters as that we use for D'AlemBERT.

FreEM$_{LPM}$ provides a standard split (train, dev, test), however it also proposes an evaluation on a *out-of-domain* subcorpus that is not contained in the standard split and that is separated by century (from the 16th to the 20th century) and that also contains both the *Normalized* and *Original* versions of the texts for the 16th, 17th and 18th centuries. The idea of this out-of-domain evaluation corpus is to have a fine-grained evaluation of the models to better assess their performance in all the different types of text that one might encounter when working with Early Modern French data.

Original							Normalized or Contemporary						
Model	16	17	18	19	20	Avg	Model	16	17	18	19	20	Avg
Drama							*Drama*						
Pie Extended	*90.34*	*94.47*	*94.64*	-	-	*93.15*	Pie Extended	*93.69*	*95.75*	*95.61*	*95.03*	*93.71*	*94.76*
CamemBERT	87.06	89.01	90.92	-	-	89.00	CamemBERT	90.18	91.51	91.37	91.13	91.42	91.12
D'AlemBERT	**94.17**	**96.59**	**96.28**	-	-	**95.68**	D'AlemBERT	**96.25**	**96.97**	**96.80**	**96.25**	**95.00**	**96.25**
Varia							*Varia*						
Pie Extended	*89.85*	*93.44*	*95.98*	-	-	*93.09*	Pie Extended	*92.52*	*94.81*	*95.98*	*92.24*	*94.03*	*93.94*
CamemBERT	86.90	88.85	92.85	-	-	89.53	CamemBERT	89.79	90.69	93.06	90.54	89.78	93.94
D'AlemBERT	**93.86**	**95.73**	**96.95**	-	-	**95.51**	D'AlemBERT	**94.52**	**96.64**	**96.88**	**94.90**	**95.30**	**95.65**
Both							*Both*						
Pie Extended	*90.08*	*93.95*	*95.33*	-	-	*93.12*	Pie Extended	*93.08*	*95.28*	*95.80*	*93.65*	*93.87*	*94.35*
CamemBERT	86.98	88.93	91.89	-	-	89.27	CamemBERT	89.99	91.10	92.22	90.84	90.60	92.53
D'AlemBERT	**94.02**	**96.16**	**96.62**	-	-	**95.60**	D'AlemBERT	**95.39**	**96.81**	**96.84**	**95.58**	**95.15**	**95.95**

Table 15.1: Comparison between D'AlemBERT, CamemBERT and Pie Extended performance on the test set, out-of-domian data of FreEM$_{LPM}$. Best scores in **bold** sencond best in *italics*.

[3] https://github.com/flairNLP/flair

Following the approach of Clérice (2020), we report the scores obtained on the out-of-domain testing dataset of FreEM$_{LPM}$ in Table 15.1. We use the scores previously reported by Clérice (2020) using *Pie Extended*, a stacked BiLSTM-CRF model, as our first baseline as well as the fine-tuned CamemBERT that serves as a second baseline as well as a rough estimation of how much knowledge can D'AlemBERT transfer from the FreEM$_{max}$ corpus into this task.

We can see that D'AlemBERT consistently outperforms Pie Extended and CamemBERT in both the normalized and original versions of our out-of-domain testing data and for all different periods by a considerable margin. Furthermore, we can also see that on average the difference in score between D'AlemBERT and Pie Extended is greater for the original split than the normalized one. This suggests that D'AlemBERT can generalize more effectively to non-normalized data than the more traditional architecture used by Pie Extended. Moreover, we can also see that the difference in scores is also greater for the 16th c. and 17th c. data. This is interesting, especially for the 16th c, because, as we can see in Figure 10.2, this is the least represented period in the FreEM$_{max}$ corpus. This result actually suggests that D'AlemBERT might be able to do effective transfer learning from the 18th c., 19th c. and 20th c. data to the 16th c. and 17th c. data.

As for CamemBERT, we can see that it consistently scores lower than both D'AlemBERT and Pie Extended. Moreover, we can see that it struggles particularly with the non-normalized data of the 16th c., 17th c. and 18th c.. This results clearly shows that CamemBERT cannot easily generalize to these earlier states of languages, or at least not with the quantity of data found in the training set of FreEM$_{LPM}$. These results also show the impressive capacity of D'AlemBERT of quickly generalizing to diverse set of states of language, as well as its capacity to transfer knowledge from the FreEM$_{max}$ corpus into this task. The obtained results are also a testament to the importance of the pre-training data, specially taking in account that the pre-training set of CamemBERT is more than 100 times bigger than that of D'AlemBERT.

15.2.2 Named Entity Recognition

Model	Precision	Recall	F1-Score
LSTM-CRF	0.8640	0.8533	0.8586
CamemBERT	*0.9303*	*0.9309*	*0.9306*
D'AlemBERT	**0.9329**	**0.9323**	**0.9326**

Table 15.2: Comparison between D'AlemBERT, CamemBERT and an LSTM-CRF-based model performance on the test set of FreEM$_{NER}$. Best scores in **bold** sencond best in *italics*.

Now we fine-tune D'AlemBERT on NER with the FreEM$_{NER}$ corpus. We use once again the `flair` framework[4] for sequence tagging (Akbik et al., 2019) and we follow the same approach as Schweter and Akbik (2020) with the exact same modifications as in the previous subsection. We also fine-tune CamemBERT using the exact same hyperparameters as that we use for D'AlemBERT. For a baseline we use the BiLSTM-CRF implementation provided by the `flair` library, and we couple it with character embeddings as well as the Common Crawl-based FastText embeddings (Grave et al., 2018) originally trained by Facebook. Results for all models can be seen in Table 15.2.

In contrast to our POS tagging experiments, here we see D'AlemBERT getting marginally better scores than D'AlemBERT, we believe that this is due to the striking size of FreEM$_{NER}$ which has more than 5 million annotated tokens, that is, we believe that in this case CamemBERT has enough training data in order to properly fine-tune to this task in Early Modern French and in particular to potentially overcome the poor representations given by the SentencePiece (Kudo and Richardson, 2018) trained on Contemporary French for the out-of-vocabulary words found in the Early Modern French data.[5] We believe that to a certain extent, given the size of FreEM$_{NER}$, CamemBERT might be "*forgetting*" its pre-training contemporary data and "*re-learning*" the Early Modern French data in FreEM$_{NER}$.

In any case, the results obtained here by D'AlemBERT are on par with the state-of-the-art NER models for Contemporary English (Wang et al., 2021) while using a much simpler architecture. The results obtained by both Transformer-based models largely outperform those obtained by our LSTM-CRF based baseline, which shows how well the Transformer-based models respond to large quantities of annotated data. We report the results by entity type on the appendix section G.2.

15.3 Conclusion

In this chapter we showed that it is possible to successfully train a Transformer-based language model for Early Modern French from scratch with even less data than originally shown in Chapter 11 and in (Martin et al., 2020). Moreover, with our POS tagging evaluation we were able to observe some form of transfer learning and generalization across multiple states of the language corresponding to different periods of time, while in our experiments in named entity recognition we observed the type of performance one can get when given a big enough annotated corpus, even when the models are not particularly fine-tuned to the specific period of time of the annotated data.

We believe that D'AlemBERT will be of use not only to the BASNUM project, but also to all digital humanists and linguists interested in Early Modern French. For our

[4] https://github.com/flairNLP/flair

[5] We observe that SentencePiece tends to split OOV words by characters which might not be ideal for sequence-tagging tasks, specially for NER.

future work, we hope that we will be able to study the application of our D'AlemBERT model to other NLP tasks such as text normalization or even document structuring, where we hope to more extensively study the transfer learning capabilities of our approach.

Part V

Conclusions and Perspectives

16 Conclusions and Perspectives

In which we present the conclusions of this Ph.D. thesis and outline future perspectives and research directions that might develop in the coming years.

During the course of this Ph.D. thesis we developed 3 versions of a large multilingual corpus, we participated in the development of one balanced corpus for Contemporary French, the curation of a corpus for Medieval French and another for Early Modern French. With these corpora, we developed contextualized ELMo representations (Peters et al., 2018) for six languages: Bulgarian, Catalan, Danish, Finnish, French and Indonesian. We also participated in the development of three RoBERTa-based (Liu et al., 2019) models for Contemporary, Medieval and Early Modern French.

We chose to focus on the development of data for the pre-training of language models rather than on the architectures themselves. This approach proved to be very effective as we were able to establish new states of the art for a wide range of tasks in natural language processing for five of the six aforementioned languages as well as for Contemporary, Medieval and Early Modern French. Furthermore, we were able to determine, not only that these contextualized language models are extremely sensitive to pre-training data quality, heterogeneity and balance, but we also showed that these three features were better predictors of the pre-trained models' performance in downstream tasks than the pre-training data size itself. In fact, we were able to determine that the importance of pre-training dataset size had been largely overestimated (Martin et al., 2020) as we were able to repeatedly show that one can pre-train these architectures with quite modestly size corpora.

We consider that we have reached and far exceeded all the initial objectives set by the BASNUM project for this thesis, by producing models and automatic annotation systems that will make it possible to enrich not only the *Dictionnaire Universel* of Basnage but also any other type of document in French from any possible time period. Moreover, due to recent developments by the *GROBID* (Grobid contributors, 2008 — 2018) and *DeLFT* (DeLFT contributors, 2018) authors, it is now possible to easily plug all of the contextualized language models that we have developed to their document structuring pipeline, we hope that these will quickly allow the members of the BASNUM project to use our French models and in particular, our D'AlemBERT model with *GROBID-dictionaries* (Khemakhem et al., 2017, 2018), hoping that they

can largely improve the initial annotations of the macrostructure already performed by Khemakhem (2020) in just a few lines of code.

We hope that the resources we have produced here will be of great use to researchers in both natural language processing and digital humanities. In particular, we hope that we will be able to continue working in improving our OSCAR project, which grew from a side-project intended to gather resources for pre-training language models in French, to a fully fledge Open Source project with a thriving community.[1,2,3] The first version of the OSCAR corpus, OSCAR 2019, now has more than 14 thousand downloads[4] and has been used independent studies to train monolingual and multilingual language models in more than 18 different languages (Antoun et al., 2021; Kakwani et al., 2020; Wilie et al., 2020; Chan et al., 2020; Koutsikakis et al., 2020; Martin et al., 2020; Chriqui and Yahav, 2021; Seker et al., 2021; Delobelle et al., 2020; Dumitrescu et al., 2020; Masala et al., 2020) making it possible to NLP researchers with all different backgrounds and interests, to access some of the latest developments in state-of-the-art NLP.

However, we do acknowledge that OSCAR is far from a perfect corpus, and many of the concerns expressed by Caswell et al. (2020); Kreutzer et al. (2022) remain to be addressed. This is a direction of research that we would like to explore in the future, so that we can provide the OSCAR community with an ever-evolving, up-to-date, properly classified, high-quality and even annotated corpus.

Finally, we would like to continue our research in NLP for digital humanities, by trying to apply our methods to other historical languages, we would additionally like to study and research the latest developments in tokenization agnostic architectures (Clark et al., 2021; Xue et al., 2021) that would be of great use when one has to model historical language that exhibit free word order and non-standardized orthography. In direct relation to the BASNUM project, we would also like to study the new sparse attention architectures (Xiong et al., 2021; Beltagy et al., 2020), that are capable of handling large contextual windows and that might be of great use for tasks like document structuring were one has to deal with long sequences of text, allowing us to develop end-to-end architectures for document structuring without the need to go through dedicated document processing pipelines such as *GROBID* (Grobid contributors, 2008 — 2018) or *DeLFT* (DeLFT contributors, 2018).

[1] https://oscar-corpus.com.
[2] https://discord.com/invite/4JNg9FTar4.
[3] https://twitter.com/oscarnlp.
[4] https://huggingface.co/datasets/oscar.

Part VI

Appendices

A Goclassy: an Asynchronous Language Classification Pipeline for Common Crawl

Language	Size		Words		Language	Size		Words	
	Orig	Dedup	Orig	Dedup		Orig	Dedup	Orig	Dedup
Afrikaans	241M	163M	43,482,801	29,533,437	Lower Sorbian	13K	7.1K	1,787	966
Albanian	2.3G	1.2G	374,196,110	186,856,699	Luxembourgish	29M	21M	4,403,577	3,087,650
Amharic	360M	206M	28,301,601	16,086,628	Macedonian	2.1G	1.2G	189,289,873	102,849,595
Arabic	82G	32G	8,117,162,828	3,171,221,354	Maithili	317K	11K	69,161	874
Aragonese	1.3M	801K	52,896	45,669	Malagasy	21M	13M	3,068,360	1,872,044
Armenian	3.7G	1.5G	273,919,388	110,196,043	Malay	111M	42M	16,696,882	6,045,753
Assamese	113M	71M	6,956,663	4,366,570	Malayalam	4.9G	2.5G	189,534,472	95,892,551
Asturian	2.4M	2.0M	381,005	325,237	Maltese	24M	17M	2,995,654	2,163,358
Avaric	409K	324K	24,720	19,478	Marathi	2.7G	1.4G	162,609,404	82,130,803
Azerbaijani	2.8G	1.5G	322,641,710	167,742,296	Mazanderani	691K	602K	73,870	64,481
Bashkir	128M	90M	9,796,764	6,922,589	Minangkabau	608K	310K	5,682	4,825
Basque	848M	342M	120,456,652	45,359,710	Mingrelian	5.8M	4.4M	299,098	228,629
Bavarian	503	503	399	399	Mirandese	1.2K	1.1K	171	152
Belarusian	1.8G	1.1G	144,579,630	83,499,037	Modern Greek	62G	27G	5,479,180,137	2,412,419,435
Bengali	11G	5.8G	623,575,733	363,766,143	Mongolian	2.2G	838M	181,307,167	68,362,013
Bihari	110K	34K	8,848	2,875	Nahuatl languages	12K	11K	1,234	1,193
Bishnupriya	4.1M	1.7M	198,286	96,940	Neapolitan	17K	13K	5,282	4,147
Bosnian	447K	116K	106,448	20,485	Nepali	1.8G	1.2G	107,448,208	71,628,317
Breton	29M	16M	5,013,241	2,890,384	Newari	5.5M	4.1M	564,697	288,995
Bulgarian	32G	14G	2,947,648,106	1,268,114,977	Northern Frisian	4.4K	4.4K	1,516	1,516
Burmese	1.9G	1.1G	56,111,184	30,102,173	Northern Luri	76K	63K	8,022	6,740
Catalan	8.0G	4.3G	1,360,212,450	729,333,440	Norwegian	8.0G	4.7G	1,344,326,388	804,894,377
Cebuano	39M	24M	6,603,567	3,675,024	Norwegian Nynorsk	85M	54M	14,764,980	9,435,139
Central Bikol	885	885	312	312	Occitan	5.8M	3.7M	750,301	512,678
Central Khmer	1.1G	581M	20,690,610	10,082,245	Oriya	248M	188M	14,938,567	11,321,740
Central Kurdish	487M	226M	48,478,334	18,726,721	Ossetian	13M	11M	1,031,268	878,765
Chavacano	520	520	130	130	Pampanga	760	304	130	52
Chechen	8.3M	6.7M	711,051	568,146	Panjabi	763M	460M	61,847,806	37,555,835
Chinese	508G	249G	14,986,424,850	6,350,215,113	Persian	79G	38G	9,096,554,121	4,363,505,319
Chuvash	39M	26M	3,041,614	2,054,810	Piemontese	2.1M	1.9M	362,013	337,246
Cornish	44K	14K	8,329	2,704	Polish	109G	47G	15,277,255,137	6,708,709,674
Croatian	226M	110M	34,232,765	16,727,640	Portuguese	124G	64G	20,641,903,898	10,751,156,918
Czech	53G	24G	7,715,977,441	3,540,997,509	Pushto	361M	242M	46,559,441	31,347,348
Danish	16G	9.5G	2,637,463,889	1,620,091,317	Quechua	78K	67K	10,186	8,691
Dhivehi	126M	79M	7,559,472	4,726,660	Romanian	25G	11G	3,984,317,058	1,741,794,069
Dimli	146	146	19	19	Romansh	7.4K	6.5K	1,093	960
Dutch	78G	39G	13,020,136,373	6,598,786,137	Russia Buriat	13K	11K	963	809
Eastern Mari	7.2M	6.0M	565,992	469,297	Russian	1.2T	568G	92,522,407,837	46,692,691,520
Egyptian Arabic	66M	33M	7,305,151	3,659,419	Sanskrit	93M	37M	4,331,569	1,713,930
Emilian-Romagnol	25K	24K	6,376	6,121	Scottish Gaelic	1.9M	1.3M	310,689	207,110
English	2.3T	1.2T	418,187,793,408	215,841,256,971	Serbian	3.9G	2.2G	364,395,411	207,561,168
Erzya	1.4K	1.2K	90	78	Serbo-Croatian	25M	5.8M	5,292,184	1,040,573
Esperanto	299M	228M	48,486,161	37,324,446	Sicilian	3.3K	2.8K	554	468
Estonian	4.8G	2.3G	643,163,730	309,931,463	Sindhi	347M	263M	43,530,158	33,028,015
Finnish	27G	13G	3,196,666,419	1,597,855,468	Sinhala	1.4G	802M	93,053,465	50,864,857
French	282G	138G	46,896,036,417	23,206,776,649	Slovak	9.1G	4.5G	1,322,247,763	656,346,179
Galician	620M	384M	102,011,291	63,600,602	Slovenian	2.5G	1.3G	387,399,700	193,926,684
Georgian	3.6G	1.9G	171,950,621	91,569,739	Somali	61K	16K	1,202	472
German	308G	145G	44,878,908,446	21,529,164,172	South Azerbaijani	27M	19M	2,175,054	1,528,709
Goan Konkani	2.2M	1.8M	124,277	102,306	Spanish	278G	149G	47,545,122,279	25,928,290,729
Guarani	36K	24K	7,382	4,680	Sundanese	211K	141K	30,321	20,278
Gujarati	1.1G	722M	72,045,701	50,023,432	Swahili	13M	8.1M	2,211,927	1,376,963
Haitian	3.9K	3.3K	1,014	832	Swedish	44G	25G	7,155,994,312	4,106,120,608
Hebrew	20G	9.8G	2,067,753,528	1,032,018,056	Tagalog	573M	407M	98,949,299	70,121,601
Hindi	17G	8.9G	1,372,234,782	745,774,934	Tajik	379M	249M	31,758,142	21,029,893
Hungarian	40G	18G	5,163,936,345	2,339,127,555	Tamil	9.3G	5.1G	420,537,132	226,013,330
Icelandic	1.5G	846M	219,900,094	129,818,331	Tatar	670M	305M	51,034,893	23,825,695
Ido	147K	130K	25,702	22,773	Telugu	2.5G	1.6G	123,711,517	79,094,167
Iloko	874K	636K	142,942	105,564	Thai	36G	16G	951,743,087	368,965,202
Indonesian	30G	16G	4,574,692,265	2,394,957,629	Tibetan	187M	138M	1,483,589	936,556
Interlingua	662K	360K	180,231	100,019	Tosk Albanian	5.0M	2.8M	841,750	459,001
Interlingue	24K	1.6K	5,352	602	Turkish	60G	27G	7,577,388,700	3,365,734,289
Irish	88M	60M	14,483,593	10,017,303	Turkmen	11M	6.8M	1,113,869	752,326
Italian	137G	69G	22,248,707,341	11,250,012,896	Tuvinian	12K	7.9K	759	540
Japanese	216G	106G	4,962,979,182	1,123,067,063	Uighur	122M	83M	8,657,141	5,852,225
Javanese	659K	583K	104,896	86,654	Ukrainian	53G	28G	4,204,381,276	2,252,380,351
Kalmyk	113K	112K	10,277	10,155	Upper Sorbian	4.2M	1.8M	545,351	236,867
Kannada	1.7G	1.1G	81,186,863	49,343,462	Urdu	2.7G	1.7G	331,817,982	218,030,228
Karachay-Balkar	2.6M	2.3M	185,436	166,496	Uzbek	21M	12M	2,450,256	1,381,644
Kazakh	2.7G	1.5G	191,126,469	108,388,743	Venetian	18K	17K	3,492	3,199

Language	Size		Words		Language	Size		Words	
	Orig	Dedup	Orig	Dedup		Orig	Dedup	Orig	Dedup
Kirghiz	600M	388M	44,194,823	28,982,620	Vietnamese	68G	32G	12,036,845,359	5,577,159,843
Komi	2.3M	1.2M	201,404	95,243	Volapük	2.0M	2.0M	321,121	318,568
Korean	24G	12G	2,368,765,142	1,120,375,149	Walloon	273K	203K	50,720	37,543
Kurdish	94M	60M	15,561,003	9,946,440	Waray	2.5M	2.2M	397,315	336,311
Lao	174M	114M	4,133,311	2,583,342	Welsh	213M	133M	37,422,441	23,574,673
Latin	26M	8.3M	4,122,201	1,328,038	Western Frisian	35M	26M	5,691,077	4,223,816
Latvian	4.0G	1.8G	520,761,977	236,428,905	Western Mari	1.2M	1.1M	93,338	87,780
Lezghian	3.3M	3.0M	247,646	224,871	Western Panjabi	12M	9.0M	1,426,986	1,111,112
Limburgan	29K	27K	4,730	4,283	Wu Chinese	109K	32K	11,189	4,333
Lithuanian	8.8G	3.9G	1,159,661,742	516,183,525	Yakut	42M	26M	2,547,623	1,789,174
Lojban	736K	678K	154,330	141,973	Yiddish	141M	84M	13,834,320	8,212,970
Lombard	443K	433K	75,229	73,665	Yoruba	55K	27K	8,906	3,518
Low German	18M	13M	2,906,347	2,146,417	Yue Chinese	3.7K	2.2K	186	128
Total	6.3T	3.2T	844,315,434,723	425,651,344,234					

Table A.1: Size of the OSCAR corpus by language measured in bytes and number of words. Standard UNIX human-readable notation is used for the size in byte. We define "words" as spaced separated tokens, which gives a good estimate of the size of each corpus for languages using Latin or Cyrillic alphabets, but might give a misleading size for other languages such as Chinese or Japanese.

B A First Evaluation of the OSCAR Corpus

B.1 Computational cost and carbon footprint

In light of recent concerns about the power consumption and carbon footprint of deep learning models (Schwartz et al., 2020; Bender et al., 2021) we report the power consumption and carbon footprint of our main experiments following the approach of Strubell et al. (2019). We use the training times of each model to compute both power consumption and CO_2 emissions.

In our set-up we used two different machines, each one having 4 NVIDIA GeForce GTX 1080 Ti graphic cards and 128GB of RAM, the difference between the machines being that one uses a single Intel Xeon Gold 5118 processor, while the other uses two Intel Xeon E5-2630 v4 processors. One GeForce GTX 1080 Ti card is rated at around 250 W,[1] the Xeon Gold 5118 processor is rated at 105 W,[2] while one Xeon E5-2630 v4 is rated at 85 W.[3] For the DRAM we can use the work of Desrochers et al. (2016) to estimate the total power draw of 128GB of RAM at around 13W. Having this information, we can now use the formula proposed by Strubell et al. (2019) in order to compute the total power required to train one ELMo model:

$$p_t = \frac{1.58t(cp_c + p_r + gp_g)}{1000}$$

Where c and g are the number of CPUs and GPUs respectively, p_c is the average power draw (in Watts) from all CPU sockets, p_r the average power draw from all DRAM sockets, and p_g the average power draw of a single GPU. We estimate the total power consumption by adding GPU, CPU and DRAM consumptions, and then multiplying by the *Power Usage Effectiveness* (PUE), which accounts for the additional energy required to support the compute infrastructure. We use a PUE coefficient of 1.58, the 2018 global average for data centers (Strubell et al., 2019). In table B.1 we report the training times in both hours and days, as well as the total power draw

[1] https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-1080-ti/specifications
[2] https://ark.intel.com/content/www/us/en/ark/products/120473/intel-xeon-gold-5118-processor-16-5m-cache-2-30-ghz.html
[3] https://ark.intel.com/content/www/us/en/ark/products/92981/intel-xeon-processor-e5-2630-v4-25m-cache-2-20-ghz.html

Language	Power	Hours	Days	KWh·PUE	CO_2e
OSCAR-Based ELMos					
Bulgarian	1183	515.00	21.45	962.61	49.09
Catalan	1118	199.98	8.33	353.25	18.02
Danish	1183	200.89	8.58	375.49	19.15
Finnish	1118	591.25	24.63	1044.40	53.26
Indonesian	1183	694.26	28.93	1297.67	66.18
Wikipedia-Based ELMos					
Bulgarian	1118	15.45	0.64	27.29	1.39
Catalan	1118	51.08	2.13	90.22	4.60
Danish	1118	14.56	0.61	25,72	1.31
Finnish	1118	21.79	0.91	38.49	1.96
Indonesian	1118	20.28	0.84	35.82	1.82
Total emissions					216.78

Table B.1: Average power draw (Watts), training times (in both hours and days), mean power consumption (KWh) and CO_2 emissions (kg) for each ELMo model trained.

(in Watts) of the system used to train each individual ELMo model. We use this information to compute the total power consumption of each ELMo, also reported in table B.1.

We can further estimate the CO_2 emissions in kilograms of each single model by multiplying the total power consumption by the average CO_2 emissions per kWh in France (where the models were trained). According to the RTE (Réseau de transport d'électricité / Electricity Transmission Network) the average emission per kWh were around 51g/kWh in November 2019,[4] when the models were trained. Thus the total CO_2 emissions in kg for one single model can be computed as:

$$CO_2e = 0.051p_t$$

All emissions for the ELMo models are also reported in table B.1.

We do not report the power consumption or the carbon footprint of training the UDPipe 2.0 architecture, as each model took less than 4 hours to train on a machine using a single NVIDIA Tesla V100 card. Also, this machine was shared during training time, so it would be extremely difficult to accurately estimate the power consumption of these models.

Even though it would have been interesting to replicate all our experiments and computational cost estimations with state-of-the-art fine-tuning models such as BERT, XLNet, RoBERTa or ALBERT, we recall that these transformer-based architectures are extremely costly to train, as noted by the BERT authors on the official BERT GitHub repository,[5] and are currently beyond the scope of our computational infrastructure.

[4] https://www.rte-france.com/fr/eco2mix/eco2mix-co2
[5] https://github.com/google-research/bert

However we believe that ELMo contextualized word embeddings remain a useful model that still provide an extremely good trade-off between performance to training cost, even setting new state-of-the-art scores in parsing and POS tagging for our five chosen languages, performing even better than the multilingual mBERT model.

B.2 Number of training steps for each checkpoint and each corpus

Language	1 Epoch	3 Epochs	5 Epochs	10 Epochs
Wikipedia-Based ELMos				
Bulgarian	6,268	18,804	31,340	62,680
Catalan	20,666	61,998	103,330	206,660
Danish	5,922	17,766	29,610	59,220
Finnish	8,763	26,289	43,815	87,630
Indonesian	7,891	23,673	39,455	78,910
OSCAR-Based ELMos				
Bulgarian	143,169	429,507	715,845	1,431,690
Catalan	81,156	243,468	405,780	811,560
Danish	81,156	243,468	405,780	811,560
Finnish	181,230	543,690	906,150	1,812,300
Indonesian	263,830	791,490	1,319,150	2,638,300

Table B.2: Number of training steps for each checkpoint, for the $\text{ELMo}_{\text{Wikipedia}}$ and $\text{ELMo}_{\text{OSCAR}}$ of each language.

C Quality at a Glance: An Audit of OSCAR 2019 and other Web-Crawled Datasets

Dataset	Supercode	Subcode(s)
JW300	kg	kwy
JW300	mg	tdx
JW300	qu	que, qug, qus, quw, quy, quz, qvi, qvz
JW300	sw	swc
OSCAR	ar	arz
OSCAR	az	azb
OSCAR	sh	bs, hr, sr
OSCAR	ku	ckb
OSCAR	ms	id, min
OSCAR	no	nn
OSCAR	sq	als*
OSCAR	zh	yue, wuu
WikiMatrix	ar	arz
WikiMatrix	sh	bs, hr, sr
WikiMatrix	zh	wuu

Table C.1: Situations where two language codes are represented, but one is a superset of another by the ISO standard, leading to unclarity about the data in the supercode dataset. *The als dataset is actually in gsw.

C.1 Details on Language Code Issues

Table C.1 provides a complete lists of the corpora where one code is defined as a superset of the other by the ISO standard, and in Table C.2 we provide a complete list of the language codes in JW300 which purport to be sign language but are actually unrelated high-resource languages.

Actual language	Code in JW300
cs	cse
de	gsg
el	gss
en	ase, asf, bfi, ins, psp, sfs, zib, zsl
es	aed, bvl, csf, csg, csn, csr, ecs, esn, gsm, hds, lsp, mfs, ncs, prl, pys, ssp, vsl
fi	fse
fr	fcs,fsl
hu	hsh
id	inl
it	ise
ja	jsl
ko	kvk
pl	pso
pt	bzs, mzy, psr, sgn_AO
ro	rms
ru	rsl
sk	svk
sq	sql
st	jw_ssa
zh	csl, tss

Table C.2: There are 48 languages in the JW300 corpus with language codes that correspond to sign languages, but in reality are unrelated high-resource languages (usually the most spoken language in the country of origin of the sign language). This table shows the actual language of the data corresponding to each sign language code.

Special attention needs to be given to the JW300 dataset, which, in addition to the sign languages and superset code issues, has a variety of other peculiarities. These problems seem to originate in the codes used by jw.org,[1] which were apparently not checked in the creation of the JW300 dataset. An overview is provided in Table C.3, and the following paragraphs give specifics.

Twelve languages in JW300 have codes starting in jw_, suggesting they are varieties of Javanese (ISO639-1 jw), but are instead attempts to represent language dialects for which there are no BCP-47 codes. These codes seem to have been updated in jw.org to appropriate BCP-47 private-use extensions in the form <supercode>_x_<tag>, which are provided in Table C.3. Twelve languages have codes starting in jw_, suggesting they are varieties of Javanese, but are instead mis-parsed private-use

[1]The jw.org website seems to use correct BCP-47 extensions now, however, and entering a code such as "jw_dmr" redirects to "naq_x_dmr".

Code in JW300	BCP-47 code	Actual Language Name
Incorrect private-use extensions		
hy_arevmda	hyw	Western Armenian
jw_dgr	os_x_dgr	Digor Ossetian
jw_dmr	naq_x_dmr	Damara Khoekhoe
jw_ibi	yom_x_ibi	Ibinda Kongo
jw_paa	pap_x_paa	Papiamento (Aruba)
jw_qcs	qxl	Salasaca Highland Kichwa
jw_rmg	rmn_x_rmg	Greek Romani (South)
jw_rmv	rmy_x_rmv	Vlax Romani, Russia
jw_spl	nso_x_spl	Sepulana
jw_ssa	st_ZA	Sesotho (South Africa)
jw_tpo	pt_PT	Portuguese (Portugal)
jw_vlc	ca_x_vlc	Catalan (Valencia)
jw_vz	skg_x_vz	Vezo Malagasy
rmy_AR	rmy_x_?	Kalderash
Equivalent codes used in place of extensions		
kmr_latn	kmr_x_rdu	Kurmanji (Caucasus)
nya	ny_x_?	Chinyanja (Zambia)
que	qu_x_?	Quechua (Ancash)
Deprecated codes		
daf	dnj/lda	Dan
ISO-693-3 used in place of ISO-693-2		
cat	ca	Catalan
gug	gn	Guarani
run	rn	Kirundi
tso_MZ	ts_MZ	Changana (Mozambique)

Table C.3: Language code issues in the JW300 datasets for 22 language varieties not covered by Tables C.1 and C.2. Private use extensions are given as they appear in `jw.org`, and specified as '?' if they are absent from `jw.org`.

extensions. Three codes appear in addition to equivalent ISO codes, making it unclear which languages they are. One language uses a deprecated ISO code. Four languages use the ISO639-3 code instead of the ISO639-2 code, and therefore are not BCP-47.

In addition to the `jw_` tags, there are two other mis-used private subtags: `hy_arevmda`, which in addition to lacking the mandatory `_x_` appears to represent standard Western Armenian (`hyw`); and `rmy_AR`, which, rather than being Romany from Argentina, is Kalderash Romany.

There are also a few anomalies where private use extensions should have been used but other methods were found to convey the distinctions. Three codes appear in addition to equivalent ISO codes, making it unclear which languages they are. Two of these are equivalencies between ISO639-2 and ISO639-3 (`nya` and `ny` are both Chichewa, `qu` and `que` are both Quechua), and one is a script equivalency (`kmr` and `kmr_latn` are both in Latin script). In these three cases the two codes do represent different languages—so a private use extension would have been appropriate.

Finally, there is the more minor issue that three languages use the ISO639-3 code instead of the ISO639-2 code, and therefore are not BCP-47.

In addition to the JW300-specific errors, Table C.4 summarizes miscellaneous errors in CCAligned and OSCAR 2019 that were detailed in Section 6.2.

Dataset	Code in Corpus	Correct Code
CCAligned	`zz`	`zza`
CCAligned	`sz`	`szl`
CCAligned	`ns`	`nso`
CCAligned	`cb`	`ckb`
CCAligned	`tz`	`ber`
CCAligned	`qa`	`shn`
CCAligned	`qd`	`kac`
CCAligned	`cx`	`ceb`
mC4	`iw`	`he`
OSCAR	`eml`	`egl`
OSCAR	`als`	`gsw`
OSCAR	`sh`	`hbs`
WikiMatrix	`sh`	`hbs`

Table C.4: Miscellaneous errors in language codes.

C.2 Complete Error Taxonomy and Instructions

In addition to the examples given in Table 6.1, raters were provided with the following verbal notes on the error codes:

- `CC`: **Correct translation, natural sentence:** It's OK if it's a sentence fragment instead of a whole sentence, as long as it is not too short (about 5 words or greater). The translation does not have to be perfect.
- `CS`: **Correct Translation, but single word or short phrase:** Also includes highly repeated short phrases, like "the cat the cat the cat the cat the cat ..."
- `CB`: **Correct translation, but boilerplate:** This can be auto-generated or formulaic content, or content that one deems "technically correct but generally not

very useful to NLP models". Unfortunately, it's often not clear what should be counted as boilerplate...do your best.

- `X`: **Incorrect translation** [for parallel sentences] both source and target are in the correct language, but they are not adequate translations.
- `WL`: **Wrong language** For short sentences, especially with proper nouns, there is often a fine line between "Wrong language" and "Not language". Do your best.
- `NL`: **Not language** At least one of source and target are not linguistic content. Any sentence consisting only of a proper noun (e.g. "Tyrone Ping") should be marked as `NL`.
- `U`: **Unknown** for sentences that need verification by a native speaker. This is an auxiliary label that is resolved in most cases.

C.3 Methodological Notes

A surprising amount of work can be done without being an expert in the languages involved. The easiest approach is simply to search the internet for the sentence, which usually results in finding the exact page the sentence came from, which in turn frequently contains clues like language codes in the URL, or a headline like *News in X language*, sometimes with references to a translated version of the same page. However, for the cases where this is insufficient, here are a few tips, tricks, and observations.

No Skills Required: Things that do not require knowledge of the language(s) in question.

1. "Not language" can usually be identified by anyone who can read the script, though there are tricky cases with proper nouns.
2. Frequently, "parallel" sentences contain different numbers in the source and target (especially autogenerated content), and are easy to disqualify.
3. Errors tend to repeat. If a word is mistranslated once, it will often be mistranslated many more times throughout a corpus, making it easy to spot.

Basic Research Required: Things that do not require knowledge of the language(s) in question but can be done with basic research.

1. If it's written in the wrong script it's considered wrong language. (Sometimes the writing system is indicated in the published corpus, e.g. `bg-Latn`, but usually the language has a "default" script defined by ISO.)

2. Some types of texts come with inherent labels or markers, such as enumerators or verse numbers.

3. When all else fails, search the internet for the whole sentence or n-grams thereof! If the whole sentence can be found, frequently the language is betrayed by the web page (the language's autonym is useful in this case).

C.4 Complete Audit Results

Table for C.5 give the complete annotation percentages for OSCAR 2019. For each annotation label, we report the ratio of the annotated sentences (of max 100 sentences) that were assigned that label by the primary annotator. Repeated annotations done for agreement measurement are not included. The `C` column aggregates all correct sub-codes (`CC`, `CS`, `CB`). We also report the total number of sentences that each dataset contains for each language and the average sentence length for the audited sentences to illustrate differences across languages. The original language codes as they are published with the datasets are maintained for the sake of consistency (but should be handled with care in future work, see Section 6.2), and those with less than 20% correct sentences are highlighted. For the complete audit results for the other 4 datasets, please refer to the original Kreutzer et al. (2022) work.

	C	CC	CS	CB	WL	NL	porn	# sentences	avg length
diq	100.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	131.00
bcl	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	1	623.00
cbk	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	1	519.00
pam	100.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2	139.00
bar	25.00%	25.00%	0.00%	0.00%	0.00%	75.00%	0.00%	4	53.50
myv	100.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	5	127.00
yue	0.00%	0.00%	0.00%	0.00%	57.14%	42.86%	0.00%	7	177.00
mwl	57.14%	57.14%	0.00%	0.00%	42.86%	0.00%	0.00%	7	141.00
frr	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	9	231.56
ht	30.00%	30.00%	0.00%	0.00%	0.00%	70.00%	0.00%	10	329.10
ie	30.00%	30.00%	0.00%	0.00%	30.00%	40.00%	0.00%	11	121.70
scn	100.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	17	155.59
tyv	96.15%	96.15%	0.00%	0.00%	0.00%	3.85%	0.00%	26	167.96
mai	79.31%	75.86%	0.00%	3.45%	20.69%	0.00%	0.00%	29	141.17
bxr	100.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	37	160.76
dsb	100.00%	97.56%	0.00%	2.44%	0.00%	0.00%	0.00%	41	155.15
so	0.00%	0.00%	0.00%	0.00%	28.57%	71.43%	0.00%	42	208.24
rm	100.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	47	137.66
nah	100.00%	96.67%	0.00%	3.33%	0.00%	0.00%	0.00%	60	164.53
nap	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	61	152.11
yo	98.46%	96.92%	0.00%	1.54%	1.54%	0.00%	0.00%	64	281.57
gn	81.48%	81.48%	0.00%	0.00%	2.47%	16.05%	0.00%	81	234.95
vec	91.36%	91.36%	0.00%	0.00%	0.00%	8.64%	0.00%	81	184.90
kw	91.57%	90.36%	0.00%	1.20%	3.61%	4.82%	0.00%	83	162.75
wuu	0.00%	0.00%	0.00%	0.00%	98.84%	1.16%	0.00%	86	157.15
eml	42.57%	42.57%	0.00%	0.00%	0.00%	57.43%	0.00%	104	177.88
bh	89.42%	21.15%	0.00%	68.27%	1.92%	8.65%	0.00%	104	137.17
min	64.00%	6.00%	0.00%	58.00%	27.00%	9.00%	0.00%	180	649.85
qu	100.00%	98.97%	0.00%	1.03%	0.00%	0.00%	0.00%	425	167.27
su	99.00%	99.00%	0.00%	0.00%	0.00%	1.00%	0.00%	676	221.00
jv	97.00%	86.00%	0.00%	11.00%	1.00%	2.00%	0.00%	2350	203.08
als	93.00%	93.00%	0.00%	0.00%	6.00%	1.00%	0.00%	7997	375.44
la	98.00%	98.00%	0.00%	0.00%	2.00%	0.00%	0.00%	33838	224.11
uz	98.00%	98.00%	0.00%	0.00%	2.00%	0.00%	0.00%	34244	369.99
nds	97.03%	95.05%	0.00%	1.98%	2.97%	0.00%	0.00%	35032	344.74
sw	98.00%	98.00%	0.00%	0.00%	0.00%	2.00%	0.00%	40066	196.70
br	100.00%	96.00%	0.00%	4.00%	0.00%	0.00%	0.00%	61941	239.56
fy	97.00%	97.00%	0.00%	0.00%	2.00%	1.00%	0.00%	67762	340.23
am	81.09%	79.10%	0.00%	1.99%	18.91%	0.00%	0.00%	287142	267.43
af	100.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	517353	339.18
eu	100.00%	98.00%	0.00%	2.00%	0.00%	0.00%	0.00%	1099498	330.93
mn	98.00%	94.00%	0.00%	4.00%	2.00%	0.00%	0.00%	1430527	309.94
te	98.99%	93.94%	1.01%	4.04%	0.00%	1.01%	1.01%	1685185	412.31
kk	100.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2719851	318.93
ca	99.00%	91.00%	0.00%	8.00%	1.00%	0.00%	0.00%	13292843	333.38
nl	98.00%	94.00%	2.00%	2.00%	2.00%	0.00%	4.00%	126067610	305.01
it	87.13%	71.29%	1.98%	13.86%	11.88%	0.99%	1.98%	210348435	393.66
zh	100.00%	97.00%	0.00%	3.00%	0.00%	0.00%	1.00%	232673578	195.60
fr	100.00%	93.00%	0.00%	7.00%	0.00%	0.00%	5.00%	461349575	306.62
es	100.00%	94.00%	0.00%	6.00%	0.00%	0.00%	3.00%	488616724	268.07
en	99.00%	96.00%	0.00%	3.00%	0.00%	1.00%	1.00%	3809525119	364.65

Table C.5: Audit results for a sample of 100 sentences from **OSCAR** for each language, compared to the number of sentences available in the dataset. If fewer than 100 sentences were available, all sentences were audited Language codes are as originally published. Length is measured in number of characters. Languages with less than 20% correct sentences are boldfaced.

D Towards a Cleaner Document-Oriented Annotated OSCAR Corpus

D.1 Carbon Footprint

We use a single machine having 192 GB of RAM and two Intel Xeon Gold 5218 processors, which is rated at 125 W,[1]. For the DRAM we can use the work of Desrochers et al. (2016) to estimate the total power draw of 192GB of RAM at around 20W. The total power draw of this setting adds up to around 270 W.

Having this information, we can now use the formula proposed by Strubell et al. (2019) in order to compute the total power required to pre-train one model from scratch:

$$p_t = \frac{1.58t(cp_c + p_r)}{1000}$$

Where c is the number of CPUs, p_c is the average power draw (in Watts) from all CPU sockets and p_r the average power draw from all DRAM sockets. We estimate the total power consumption by adding CPU and DRAM consumption, and then multiplying by the *Power Usage Effectiveness* (PUE), which accounts for the additional energy required to support the compute infrastructure. We use a PUE coefficient of 1.58, the 2018 global average for data centers (Strubell et al., 2019). The total time to generate OSCAR 22.01 in this infrastructure was of 42.6 hours. We use this information to compute the total power consumption of the OSCAR generation, which amounts to 0.4266 kWh.

We can further estimate the CO_2 emissions in kilograms of the OSCAR generation by multiplying the total power consumption by the average CO_2 emissions per kWh in our region which were 38.64g/kWh in average between the 3rd and the 5th of January 2022[2], the exact time at which the generation was run. Thus the total CO_2 emissions in kg for one single model can be computed as:

$$CO_2e = 0.03864p_t$$

Thus total CO_2 emissions amount to 0.01648kg or 16.48g.

[1] Intel Xeon Gold 5218 specification
[2] Rte - éCO_2mix.

D.2 Language Table

Language	Size	Documents	Words	Language	Size	Documents	Words
Afrikaans	47.0 MB	12,393	6,227,310	Luxembourgish	15.8 MB	5,108	1,545,946
Tosk Albanian	363.6 kB	139	37,381	Lezghian	375.5 kB	124	19,250
Amharic	461.0 MB	37,513	30,481,153	Limburgish	1.4 kB	2	41
Aragonese	10.6 kB	12	51	Lombard	2.6 kB	2	225
Arabic	84.2 GB	8,718,929	6,103,711,887	Lao	337.1 MB	28,914	6,682,982
Egyptian Arabic	2.8 MB	1,256	176,096	Lithuanian	20.0 GB	2,303,070	1,712,802,056
Assamese	221.2 MB	17,084	11,109,557	Latvian	8.2 GB	1,032,987	707,361,898
Asturian	73.6 kB	77	3,919	Maithili	21.6 kB	23	483
Avaric	18.6 kB	14	582	Malagasy	57.3 MB	3,028	7,279,056
Azerbaijani	3.5 GB	491,847	291,927,692	Eastern Mari	11.3 MB	1,612	641,525
South Azerbaijani	14.1 MB	5,381	693,746	Minangkabau	6.0 MB	585	614,613
Bashkir	95.5 MB	11,198	5,418,474	Macedonian	3.6 GB	341,775	244,058,579
Belarusian	1.8 GB	180,046	107,227,860	Malayalam	4.1 GB	250,972	137,831,247
Bulgarian	35.1 GB	2,887,115	2,405,981,285	Mongolian	2.8 GB	237,719	176,405,432
Bihari languages	24.2 kB	27	569	Marathi	3.3 GB	250,376	160,179,233
Bangla	15.1 GB	1,171,501	751,877,226	Western Mari	743.5 kB	155	43,916
Tibetan	234.5 MB	18,683	2,286,269	Malay	5.3 MB	5,228	217,818
Bishnupriya	2.0 MB	271	98,419	Maltese	2.5 MB	2,208	118,190
Breton	33.7 MB	16,119	3,111,619	Multilingual	12.1 GB	1,210,685	936,187,711
Bosnian	10.3 kB	10	422	Burmese	1.9 GB	158,733	44,835,970
Russia Buriat	32.9 kB	39	785	Mazanderani	128.2 kB	76	7,337
Catalan	13.9 GB	2,627,307	1,508,919,864	Nahuatl languages	8.7 kB	12	179
Chechen	14.0 MB	4,086	798,766	Low German	9.0 MB	1,938	1,012,561
Cebuano	44.6 MB	5,742	5,253,785	Nepali	3.7 GB	391,947	177,885,116
Central Kurdish	716.4 MB	84,950	43,913,025	Newari	5.7 MB	1,134	273,837
Czech	58.6 GB	10,381,916	5,452,724,456	Dutch	114.0 GB	20,206,532	12,329,127,151
Chuvash	41.8 MB	4,750	2,465,782	Norwegian Nynorsk	6.8 MB	5,835	459,183
Welsh	409.3 MB	90,378	49,488,495	Norwegian	2.8 GB	973,188	279,182,902
Danish	12.6 GB	2,265,479	1,454,439,292	Occitan	2.1 MB	373	31,061
German	496.7 GB	70,075,424	46,826,676,844	Odia	487.9 MB	52,942	23,755,902
Dimli (individual language)	706 Bytes	1	19	Ossetic	13.9 MB	3,560	800,430
Lower Sorbian	707 Bytes	1	17	Punjabi	1.1 GB	68,094	70,068,604
Divehi	217.2 MB	24,067	10,112,205	Polish	139.0 GB	19,301,137	12,584,498,906
Greek	78.3 GB	6,738,546	5,031,242,803	Piedmontese	1.7 MB	698	188,270
Emiliano-Romagnolo.	901 Bytes	1	53	Western Panjabi	46.7 MB	6,790	4,060,419
English	3.2 TB	431,992,659	377,376,402,775	Pashto	490.3 MB	50,312	46,293,249
Esperanto	558.3 MB	111,932	58,416,628	Portuguese	170.3 GB	23,735,707	18,441,864,893
Spanish	381.9 GB	51,386,247	42,829,835,316	Quechua	744 Bytes	1	14
Estonian	9.2 GB	1,362,524	820,975,443	Romanian	49.2 GB	4,624,764	5,261,803,995
Basque	1.1 GB	233,658	97,092,942	Russian	1.1 TB	76,060,844	62,811,122,663
Persian	77.4 GB	7,665,871	6,430,164,396	Sanskrit	136.0 MB	4,472	5,671,369
Finnish	37.8 GB	4,948,961	2,900,615,928	Sakha	65.6 MB	6,284	3,473,813
French	382.2 GB	52,037,098	41,713,990,658	Sicilian	1.5 kB	2	50
Western Frisian	75.3 MB	21,946	6,357,929	Sindhi	117.1 MB	15,516	10,685,611
Irish	45.6 MB	12,233	4,877,850	Serbian (Latin)	931.8 kB	738	92,875
Scottish Gaelic	137.7 kB	136	7,769	Sinhala	2.0 GB	108,593	113,179,741
Galician	255.2 MB	88,803	27,051,212	Slovak	16.5 GB	2,409,555	1,619,121,944
Guarani	9.0 kB	10	374	Slovenian	1.2 GB	351,894	118,400,246
Goan Konkani	787.2 kB	46	38,831	Somali	2.1 kB	3	109
Gujarati	4.8 GB	136,467	301,170,777	Albanian	3.0 GB	437,287	326,325,149
Hebrew	30.3 GB	3,132,396	2,249,377,984	Serbian	6.9 GB	577,472	482,932,670
Hindi	23.3 GB	1,529,907	1,534,799,198	Sundanese	5.0 MB	263	547,145
Croatian	11.2 MB	11,462	505,369	Swedish	48.0 GB	7,541,278	5,078,331,128
Upper Sorbian	132.8 kB	110	8,825	Swahili	1.3 MB	462	123,050
Hungarian	53.9 GB	6,866,062	4,598,787,907	Tamil	11.4 GB	556,772	452,343,748
Armenian	4.7 GB	379,267	268,031,270	Telugu	3.4 GB	249,756	137,752,065
Interlingua	40.2 kB	6	10,125	Tajik	870.9 MB	46,366	56,627,727
Indonesian	17.4 GB	2,244,622	1,984,195,207	Thai	66.1 GB	5,030,254	1,626,779,846
Iloko	97.9 kB	75	8,592	Turkmen	4.4 MB	2,485	276,632
Ido	77.3 kB	105	2,690	Filipino	646.5 MB	70,394	81,881,278
Icelandic	2.0 GB	396,183	210,365,124	Turkish	75.1 GB	10,826,031	6,421,221,358
Italian	229.3 GB	28,502,092	24,294,684,830	Tatar	915.3 MB	76,398	51,875,265
Japanese	258.7 GB	36,328,931	5,592,948,356	Uyghur	201.9 MB	18,556	11,240,889
Lojban	1.9 MB	570	260,542	Ukrainian	48.8 GB	4,558,214	2,879,585,992
Javanese	152.7 kB	70	10,441	Urdu	3.4 GB	336,994	332,816,354
Georgian	7.1 GB	488,588	281,430,479	Uzbek	19.9 MB	9,526	1,370,842
Kazakh	2.9 GB	261,085	157,267,307	Vietnamese	98.9 GB	9,587,233	12,283,185,482
Khmer	1.9 GB	121,910	30,564,131	Volapük	825.9 kB	661	57,039
Kannada	2.6 GB	150,850	108,450,571	Walloon	105.7 kB	138	4,386
Korean	51.8 GB	5,881,481	3,854,968,649	Waray	7.6 MB	933	830,872
Karachay-Balkar	119.6 kB	91	4,089	Wu Chinese	137.2 kB	88	3,056
Kurdish	150.3 MB	29,906	17,390,759	Kalmyk	9.3 kB	9	250
Komi	119.9 kB	127	3,335	Mingrelian	7.6 MB	2,550	253,333
Cornish	1.4 kB	2	55	Yiddish	232.5 MB	23,418	15,809,780
Kyrgyz	518.6 MB	62,244	28,028,986	Yoruba	24.7 kB	26	1,042
Latin	4.1 MB	4,397	187,446	Chinese	900.9 GB	56,524,518	23,149,203,886

Table D.1: Size of the OSCAR 22.01 corpus by language measured in bytes and number of words. Standard UNIX human-readable notation is used for the size in byte. We define "words" as spaced separated tokens, which gives a good estimate of the size of each corpus for languages using Latin or Cyrillic alphabets, but might give a misleading size for other languages such as Chinese or Japanese.

E CamemBERT

Dataset	Masking	Arch.	#Steps	GSD		Sequoia		Spoken		ParTUT		NER	NLI
				UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS	F1	Acc.
Fine-tuning													
OSCAR	Subword	Base	100k	**98.25**	92.29	99.25	93.70	96.95	79.96	97.73	**92.68**	89.23	81.18
OSCAR	Whole-word	Base	100k	98.21	92.30	99.21	94.33	96.97	80.16	**97.78**	92.65	89.11	81.92
CCNET	Subword	Base	100k	98.02	92.06	**99.26**	94.13	96.94	80.39	97.55	92.66	89.05	81.77
CCNET	Whole-word	Base	100k	98.03	**92.43**	99.18	94.26	96.98	80.89	97.46	92.33	89.27	81.92
CCNET	Whole-word	Base	500k	98.21	**92.43**	99.24	**94.60**	96.69	**80.97**	97.65	92.48	89.08	83.43
CCNET	Whole-word	Large	100k	98.01	91.09	99.23	93.65	**97.01**	80.89	97.41	92.59	**89.39**	**85.29**
Embeddings (with UDPipe Future (tagging, parsing) or LSTM+CRF (NER))													
OSCAR	Subword	Base	100k	**98.01**	90.64	**99.27**	94.26	97.15	**82.56**	**97.70**	92.70	**90.25**	-
OSCAR	Whole-word	Base	100k	97.97	90.44	99.23	93.93	97.08	81.74	97.50	92.28	89.48	-
CCNET	Subword	Base	100k	97.87	**90.78**	99.20	94.33	**97.17**	82.39	97.54	92.51	89.38	-
CCNET	Whole-word	Base	100k	97.96	90.76	99.23	**94.34**	97.04	82.09	97.39	**92.82**	89.85	-
CCNET	Whole-word	Base	500k	97.84	90.25	99.14	93.96	97.01	82.17	97.27	92.28	89.07	-
CCNET	Whole-word	Large	100k	**98.01**	90.70	99.23	94.01	97.04	82.18	97.31	92.28	88.76	-

Table E.1: Performance reported on **Test sets** for all trained models (**average** over multiple fine-tuning seeds).

In this appendix, we analyze different design choices of CamemBERT (Table E.2), namely with respect to the use of whole-word masking, the training dataset, the model size, and the number of training steps in complement with the analyses of the impact of corpus origin and size (Section 11.3). In all the ablations, all scores come from at least 4 averaged runs. For POS tagging and dependency parsing, we average the scores on the 4 treebanks. We also report all averaged test scores of our different models in Table E.1.

E.1 Impact of Whole-Word Masking

In Table E.2, we compare models trained using the traditional subword masking with whole-word masking. Whole-Word Masking positively impacts downstream performances for NLI (although only by 0.5 points of accuracy). To our surprise, this Whole-Word Masking scheme does not benefit much lower level task such as Name Entity Recognition, POS tagging and Dependency Parsing.

E.2 Impact of model size

Table E.2 compares models trained with the BASE and LARGE architectures. These models were trained with the CCNet corpus (135 GB) for practical reasons. We con-

Dataset	Masking	Arch.	#Param.	#Steps	UPOS	LAS	NER	XNLI
Masking Strategy								
OSCAR	Subword	Base	110M	100k	97.78	89.80	**91.55**	81.04
OSCAR	Whole-word	Base	110M	100k	**97.79**	**89.88**	91.44	**81.55**
Model Size								
CCNet	Whole-word	Base	110M	100k	97.67	89.46	90.13	82.22
CCNet	Whole-word	Large	335M	100k	**97.74**	**89.82**	**92.47**	**85.73**
Dataset								
CCNet	Whole-word	Base	110M	100k	97.67	89.46	90.13	**82.22**
OSCAR	Whole-word	Base	110M	100k	**97.79**	**89.88**	**91.44**	81.55
Number of Steps								
CCNet	Whole-word	Base	110M	100k	**98.04**	89.85	90.13	82.20
CCNet	Whole-word	Base	110M	500k	97.95	**90.12**	91.30	**83.04**

Table E.2: Comparing scores on the **Validation sets** of different design choices. POS tagging and parsing datasets are averaged. (average over multiple fine-tuning seeds).

firm the positive influence of larger models on the NLI and NER tasks. The LARGE architecture leads to respectively 19.7% error reduction and 23.7%. To our surprise, on POS tagging and dependency parsing, having three time more parameters doesn't lead to a significant difference compared to the BASE model. Tenney et al. (2019) and Jawahar et al. (2019) have shown that low-level syntactic capabilities are learned in lower layers of BERT while higher level semantic representations are found in upper layers of BERT. POS tagging and dependency parsing probably do not benefit from adding more layers as the lower layers of the BASE architecture already capture what is necessary to complete these tasks.

E.3 Impact of training dataset

Table E.2 compares models trained on CCNet and on OSCAR. The major difference between the two datasets is the additional filtering step of CCNet that favors Wikipedia-Like texts. The model pretrained on OSCAR gets slightly better results on POS tagging and dependency parsing, but gets a larger +1.31 improvement on NER. The CCNet model gets better performance on NLI (+0.67).

E.4 Impact of number of steps

Figure E.1 displays the evolution of downstream task performance with respect to the number of steps. All scores in this section are averages from at least 4 runs with different random seeds. For POS tagging and dependency parsing, we also average the scores on the 4 treebanks.

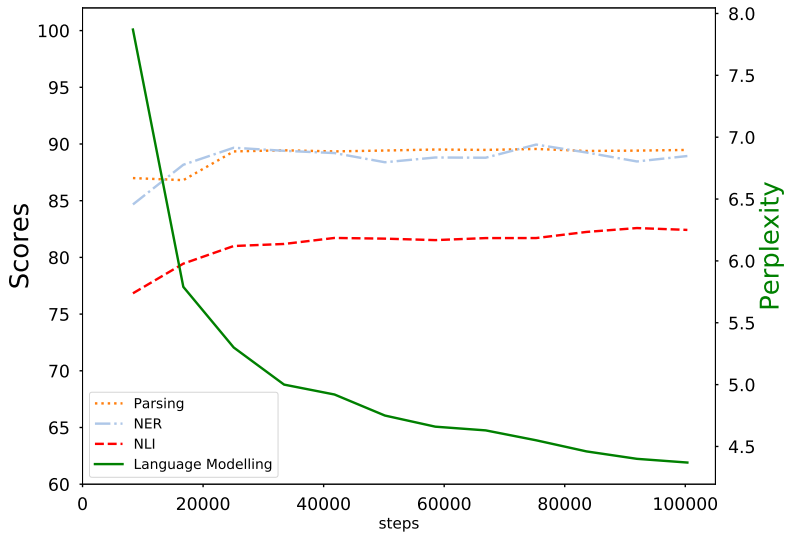


Figure E.1: Impact of number of pretraining steps on downstream performance for CamemBERT.

.

We evaluate our model at every epoch (1 epoch equals 8360 steps). We report the masked language modelling perplexity along with downstream performances. Figure E.1, suggests that the more complex the task the more impactful the number of steps is. We observe an early plateau for dependency parsing and NER at around 22k steps, while for NLI, even if the marginal improvement with regard to pretraining steps becomes smaller, the performance is still slowly increasing at 100k steps.

In Table E.2, we compare two models trained on CCNet, one for 100k steps and the other for 500k steps to evaluate the influence of the total number of steps. The model trained for 500k steps does not increase the scores much from just training for 100k steps in POS tagging and parsing. The increase is slightly higher for XNLI (+0.84).

Those results suggest that low level syntactic representation are captured early in the language model training process while it needs more steps to extract complex semantic information as needed for NLI.

F BERTrade

F.1 Collecting the Data

The following data can be downloaded directly from their website:

- Chartes de l'Aube:
  https://sites.google.com/site/achimstein/research/resources
  Extract raw text from XML files: <body>, then <s>, then <word>.

- Geste:
  https://github.com/Jean-Baptiste-Camps/Geste
  Raw text is available under /txt/norm/.

- OpenMedFr:
  https://github.com/OpenMedFr/texts
  Remove the header of each file (until *** *START*), its last line (*** *END*), paragraph breaks (#|) and folios or pages numbers.

Special permissions are required to access and use these sources:

- AND:
  https://anglo-norman.net/project-members

- BFM:
  http://bfm.ens-lyon.fr/spip.php?article19
  Raw text is available.

- Chartes Douai:
  https://www.rose.uzh.ch/docling

- MCVF: http://www.voies.uottawa.ca

- NCA:
  https://sites.google.com/site/achimstein/research/resources
  Extract raw text from the XML files: <body> then <txm:form>.

F.2 Details on the Models

F.2.1 Models Trained From Scratch

These are trained for 32 epochs in a masked language modeling task using the same parameters as RoBERTa (Liu et al., 2019) but a smaller batch size of 256 samples[1], which amounts to a magnitude of 10^5 steps. We also use a smaller vocabulary size (8192) than other works, in line with the observations of Ding et al. (2019) that learning large vocabularies on small corpora defeats the purpose of sub-word tokenization. Using a larger vocabulary size of $5{\times}10^4$ (like FlauBERT) also did not seem to bring any improvements in our preliminary experiments and made pre-training more expensive.

F.2.2 Post-training

The pretrained models we used in the post-training settings are those available in the 4.2.0 version of Huggingface Transformers (Wolf et al., 2020) and the exact handles are:

mBERT bert-base-multilingual-cased

flauBERT flaubert/flaubert_base_cased

camemBERT camembert-base

finBERT TurkuNLP/bert-base-finnish-cased-v1

The post-trained models are those with MLM heads, which we did not reset before post-training, so the post-training phase can be seen as a language transfer task for masked language modeling out of which we extract a contextual word embeddings model.

F.3 Carbon Footprint

We report the power consumption and carbon footprint of our main experiments following the approach of Strubell et al. (2019). Two different configurations were used in our experiments, one for pre-training models from scratch (Pre-train) and another one for continuing the training of existing models (Post-train).

[1] Preliminary experiments with larger batch sizes showed no significant improvement to compensate for the heavier computational load.

Model	Power (W)	# Models	Duration (h)	Consumption (kWh)	CO_2e (kg)
Pre-train	10756	11	6	11216.36	358.92
Post-train	1520	4	20	192.13	6.15
Total emissions					365.07

Table F.1: Average power draw, number of models trained, training times in hours, mean power consumption including power usage effectiveness (PUE), and CO_2 emissions; for each setting.

Pre-train: We use a cluster of 4 machines each one having 8 GPU Nvidia Tesla V100 SXM2 32 GiB, 384 GiB of RAM, and two Intel Xeon Gold 6226 processors. One Nvidia Tesla V100 card is rated at around 300 W,[2] while the Xeon Gold 6226 processor is rated at 125 W,[3]. For the DRAM we can use the work of Desrochers et al. (2016) to estimate the total power draw of 384 GiB of RAM at around 39 W. The total power draw of this setting adds up to around 10 756 W. We train 11 different models in this configuration.

Post-train: We use a single machine having 4 GPU Nvidia Tesla V100 SXM2 32 GiB, 192 GiB of RAM and two Intel Xeon Gold 6248 processors. The Xeon Gold 6248 processor is rated at 150 W,[4], and the DRAM total power draw can be estimated at around 20 W. The total power draw of this setting adds up to around 1520 W. We train 4 different models in this configuration.

Having this information, we can now use the formula proposed by Strubell et al. (2019) in order to compute the total power required for each setting:

$$p_t = \frac{1.58t(cp_c + p_r + gp_g)}{1000}$$

Where c and g are the number of CPUs and GPUs respectively, p_c is the average power draw (in W) from all CPU sockets, p_r the average power draw from all DRAM sockets, and p_g the average power draw of a single GPU. We estimate the total power consumption by adding GPU, CPU and DRAM consumption, and then multiplying by the *Power Usage Effectiveness* (PUE), which accounts for the additional energy required to support the compute infrastructure. We use a PUE coefficient of 1.58, the 2018 global average for data centers (Strubell et al., 2019). In table F.1 we report the training times in hours, as well as the total power draw (in Watts) of the system used to train the models. We use this information to compute the total power consumption of each setting, also reported in table F.1.

We can further estimate the CO_2 emissions in kilograms of each single model by multiplying the total power consumption by the average CO_2 emissions per kWh in

[2] Nvidia Tesla V100 specification
[3] Intel Xeon Gold 6226 specification
[4] Intel Xeon Gold 6248 specification

our region which were around 32 g kW^{-1} h in January 2021,[5] when the models were trained. Thus the total CO_2 emissions in kg for one single model can be computed as:

$$CO_2e = 0.032p_t$$

All emissions are also reported in table F.1.

[5] Rte - éCO_2mix.

G D'AlemBERT

G.1 Carbon Footprint

Model	Power (W)	Time (h)	(PUE·kWh)	CO^2e (kg)
Pre-train	48640	20	1537.02	46.11
Evaluation	589	1	0.93	0.03
Total CO^2e				46.14

Table G.1: Average power draw, number of models trained, training times in hours, mean power consumption including power usage effectiveness (PUE), and CO^2 emissions; for each setting.

In light of recent interest concerning the energy consumption and carbon emission of machine learning models and specifically of those of language models (Schwartz et al., 2020; Bender et al., 2021), we have decided to report the power consumption and carbon footprint of all our experiments following the approach of Strubell et al. (2019). We report the energy consumption and carbon emissions of both the pre-training of D'AlemBERT and its evaluation.

Pre-training: We use a cluster of 32 machines, each one having 4 GPU Nvidia Tesla V100 SXM2 32GiB, 192GiB of RAM, and two Intel Xeon Gold 6248 processors. One Nvidia Tesla V100 card is rated at around 300W,[1] while the Xeon Gold 6248 processor is rated at 150W.[2] For the DRAM we can use the work of Desrochers et al. (2016) to estimate the total power draw of 192GiB of RAM at around 20W. Thus, the total power draw of the pre-training adds up to around 48640W.

Evaluation: We use a single machine with a single GPU Nvidia Tesla V100 SXM2 32GiB, 384GiB of RAM and two Intel Xeon Gold 6226 processors. The Xeon Gold 6226 processor is rated at 125 W,[3] and the DRAM total power draw can be estimated at around 39W. Therefore, the total power draw of the evaluation adds up to around 589W.

[1] Nvidia Tesla V100 specification
[2] Intel Xeon Gold 6248 specification
[3] Intel Xeon Gold 6226 specification

With this information, we use the formula proposed by Strubell et al. (2019) to compute the total power required for each setting:

$$p_t = \frac{1.58t(cp_c + p_r + gp_g)}{1000}$$

Where c and g are the number of CPUs and GPUs respectively, p_c is the average power draw (in W) from all CPU sockets, p_r the average power draw from all DRAM sockets and p_g the average power draw of a single GPU. We estimate the total power consumption by adding GPU, CPU and DRAM consumption, and then multiplying by the *Power Usage Effectiveness* (PUE), which accounts for the additional energy required to support the compute infrastructure. We use a PUE coefficient of 1.58, the 2018 global average for data centers (Strubell et al., 2019). In Table G.1 we report the training times in hours, as well as the total power draw (in Watts) of the system used to train the models. We use this information to compute the total power consumption of each setting, also reported in Table G.1.

We can further estimate the CO^2 emissions in kilograms of each single model by multiplying the total power consumption by the average CO^2 emissions per kWh in our region, which were around 30g/kWh between the 30th and the 31st of December,[4] when the models were trained. Thus the total CO^2 emissions in kg for one single model can be computed as:

$$CO_2e = 0.030p_t$$

All emissions are also reported in Table G.1.

G.2 Detail Results of Experiments in NER by Entity Type

Here we show the results of each of the trained NER models by entity type.

G.3 Entity Distribution by Text in NER Data

The following diagrams show the detail of the coarse entity distribution by text in FreEM$_{NER}$.

[4] Rte - éCO²mix.

LSTM-CRF				
Entity Type	Precision	Recall	F1-Score	Support
pers	0.8808	0.8435	0.8617	2734
loc	0.8109	0.8707	0.8397	1384
amount	0.9040	0.9040	0.9040	250
time	0.9604	0.9237	0.9417	236
func	0.8872	0.8429	0.8645	140
org	0.8824	0.6122	0.7229	49
prod	0.9231	0.4444	0.6000	27
event	0.7273	0.6667	0.6957	12
micro avg	0.8640	0.8533	0.8586	4832
macro avg	0.8720	0.7635	0.8038	4832
weighted avg	0.8659	0.8533	0.8583	4832
samples avg	0.7737	0.7737	0.7737	4832

Table G.2: Results of the BiLSTM-CRF model on the test set of FreEM$_{NER}$ by entity type.

CamemBERT				
Entity Type	Precision	Recall	F1-Score	Support
pers	0.9373	0.9236	0.9304	2734
loc	0.9140	0.9371	0.9254	1384
amount	0.9840	0.9840	0.9840	250
time	0.9447	0.9407	0.9427	236
func	0.9209	0.9143	0.9176	140
org	0.8364	0.9388	0.8846	49
prod	0.7742	0.8889	0.8276	27
event	0.8333	0.8333	0.8333	12
micro avg	0.9303	0.9309	0.9306	4832
macro avg	0.8931	0.9201	0.9057	4832
weighted avg	0.9307	0.9309	0.9307	4832
samples avg	0.8856	0.8856	0.8856	4832

Table G.3: Results of CamemBERT on the test set of FreEM$_{NER}$ by entity type.

D'AlemBERT				
Entity Type	Precision	Recall	F1-Score	Support
pers	0.9355	0.9279	0.9317	2734
loc	0.9242	0.9335	0.9288	1384
amount	0.9800	0.9800	0.9800	250
time	0.9456	0.9576	0.9516	236
func	0.9333	0.9000	0.9164	140
org	0.8148	0.8980	0.8544	49
prod	0.8621	0.9259	0.8929	27
event	0.8333	0.8333	0.8333	12
micro avg	0.9329	0.9323	0.9326	4832
macro avg	0.9036	0.9195	0.9111	4832
weighted avg	0.9331	0.9323	0.9327	4832
samples avg	0.8893	0.8893	0.8893	4832

Table G.4: Results of D'AlemBERT model on the test set of FreEM$_{NER}$ by entity type.

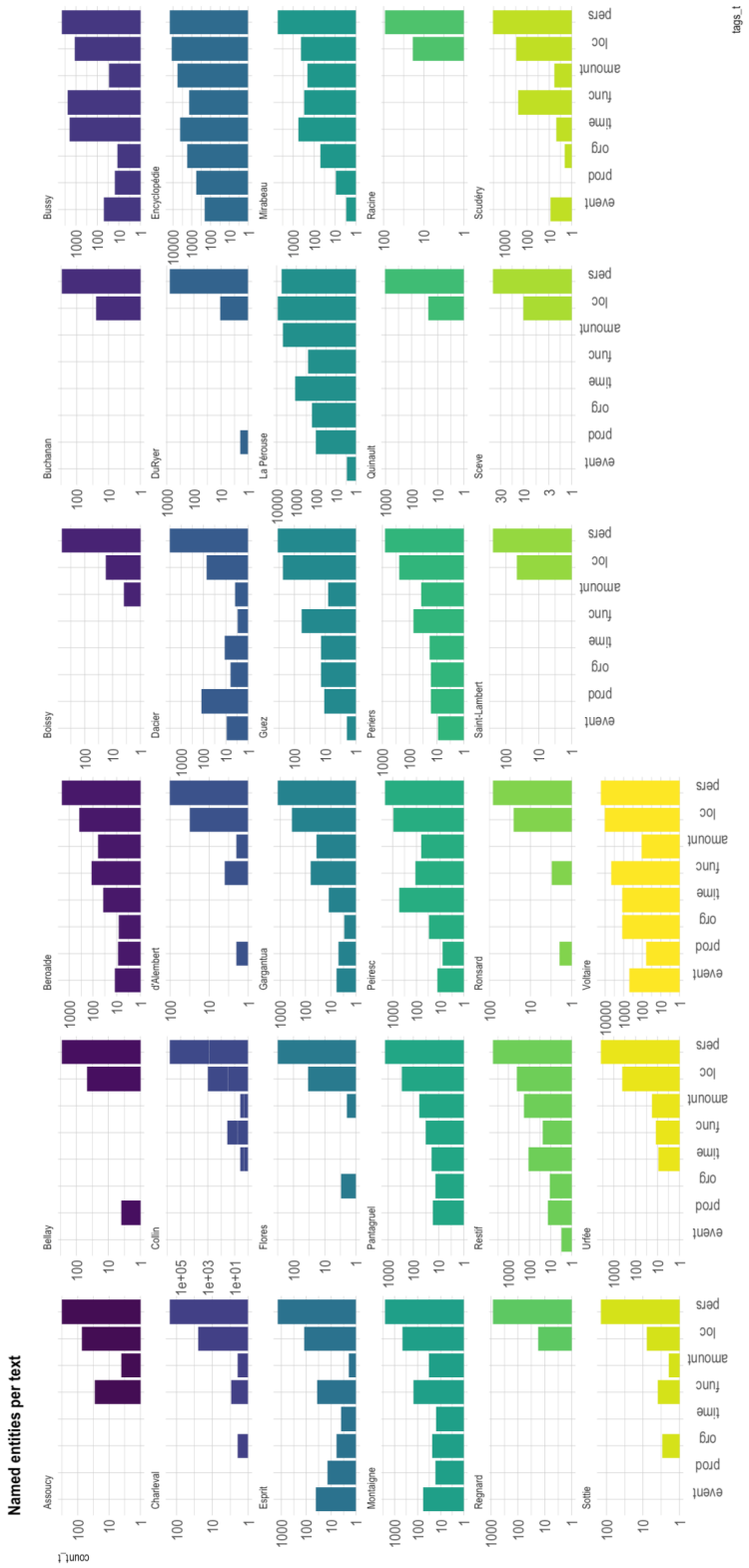


Figure G.1: Number of entities by text on a logarithmic scale.

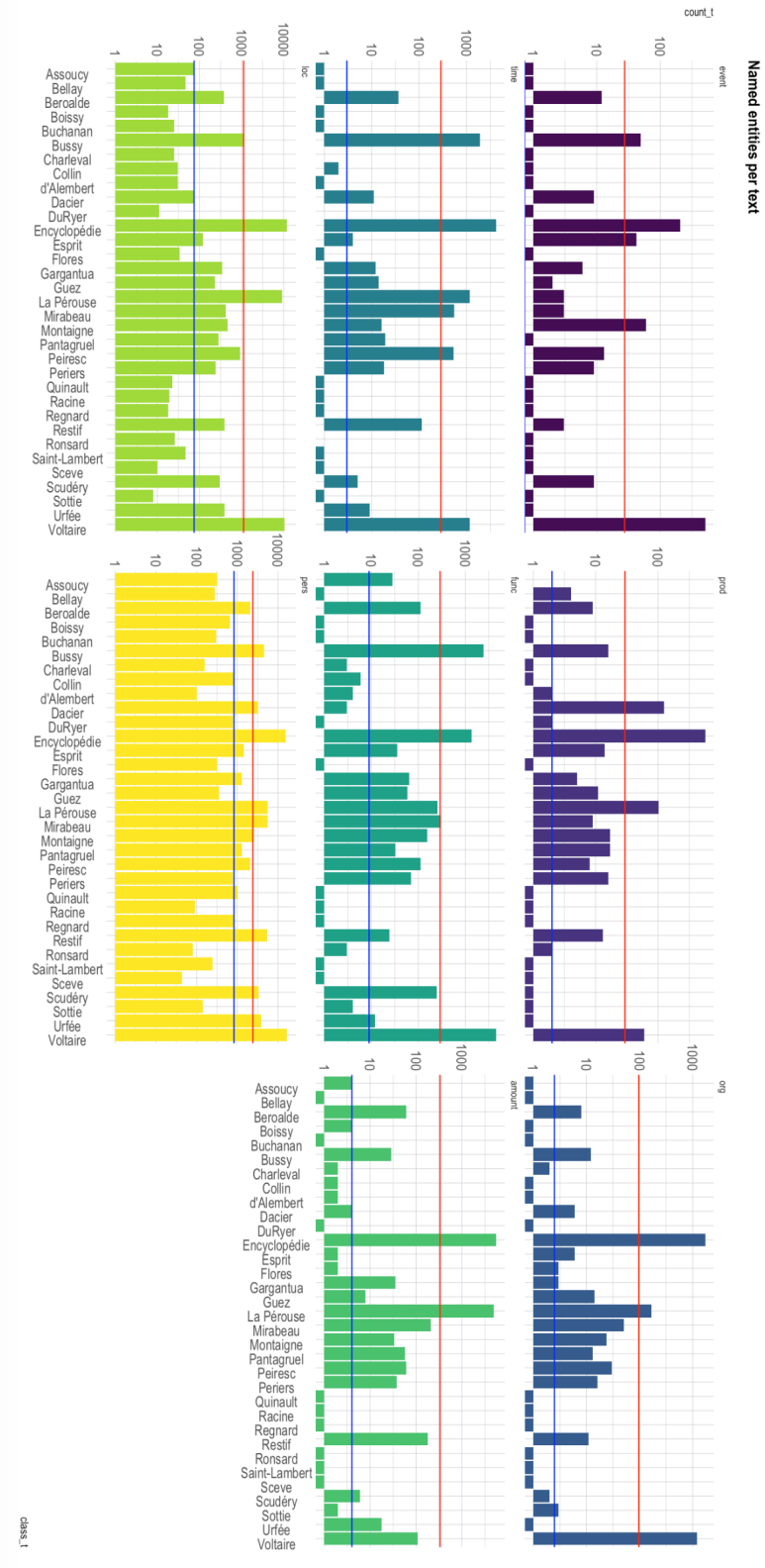


Figure G.2: Entity types by text on a logarithmic scale.

Bibliography

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. In *Proceedings of The 13th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association.

Julien Abadji, Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2021. Ungoliant: An optimized pipeline for the generation of a very large-scale multilingual web corpus. In *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-9) 2021. Limerick, 12 July 2021 (Online-Event)*, Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-9) 2021. Limerick, 12 July 2021 (Online-Event), pages 1 – 9, Mannheim. Leibniz-Institut für Deutsche Sprache.

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. *Building a Treebank for French*, pages 165–187. Springer Netherlands, Dordrecht.

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Rami Al-Rfou', Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Antonella Amatuzzi, Carine Skupien Dekens, Wendy Ayres-Bennett, Annette Gerstenberg, and Lene Schoesler. 2019. Améliorer et appliquer les outils numériques.

ressources et approches pour l'étude du changement linguistique en français préclassique et classique. In *Le français en Diachronie*, Travaux de Linguistique Romane, pages 337–364. Editions de linguistique et de philologie.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Oxford Text Archive. 2001. The york-helsinki parsed corpus of old english poetry (YCOEP). Oxford Text Archive.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

ATILF. 1998–a. Frantext démonstration.

ATILF. 1998–b. Frantext intégral.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Wendy Ayres-Bennett and Magali Seijido, editors. 2011. *Remarques et observations sur la langue française. Histoire et évolution d'un genre*. Number 1 in Histoire et évolution du français. Classiques Garnier.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven pretraining of self-attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5360–5369, Hong Kong, China. Association for Computational Linguistics.

David Bamman and Patrick J. Burns. 2020. Latin BERT: A Contextual Language Model for Classical Philology. *arXiv e-prints*, page arXiv:2009.10053.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Michael Barera. 2020. Mind the gap: Addressing structural equity and inclusion on wikipedia. https://rc.library.uta.edu/uta-ir/handle/10106/29572.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Rachel Bawden, Marie-Amélie Botalla, Kim Gerdes, and Sylvain Kahane. 2014. Correcting and validating syntactic dependency in the spoken French treebank rhapsodie. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2320–2325, Reykjavik, Iceland. European Language Resources Association (ELRA).

Frédéric Béchet, Benoît Sagot, and Rosa Stern. 2011. Coopération de méthodes statistiques et symboliques pour l'adaptation non-supervisée d'un système d'étiquetage en entités nommées (statistical and symbolic methods cooperation for the unsupervised adaptation of a named entity recognition system). In *Actes de la 18e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, pages 19–24, Montpellier, France. ATALA.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv e-prints*, page arXiv:2004.05150.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? 🦜. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Michaël Benesty. Ner algo benchmark: spacy, flair, m-bert and camembert on anonymizing french commercial legal cases [online]. 2019.

Aleksandrs Berdicevskis and Hanne Eckhoff. 2020. A diachronic treebank of Russian spanning more than a thousand years. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5251–5256, Marseille, France. European Language Resources Association.

Douglas Biber. 1993. Representativeness in Corpus Design. *Literary and Linguistic Computing*, 8(4):243–257.

Stella Biderman and Walter J. Scheirer. 2020. Pitfalls in Machine Learning Research: Reexamining the Development Cycle. *arXiv e-prints*, page arXiv:2011.02832.

Abeba Birhane and Vinay Uday Prabhu. 2021. Large image datasets: A pyrrhic win for computer vision? In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1536–1546.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Peter Blumenthal, Sascha Diwersy, Achille Falaise, Marie-Hélène Lay, Gilles Souvay, and Denis Vigier. 2017. Presto, un corpus diachronique pour le français des XVIe-XXe siècles. In *Actes de la 24ème conférence sur le Traitement Automatique des Langues Naturelles - TALN'17*. Association pour le traitement automatique des langues.

Peter Blumenthal and Denis (dir.) Vigier. 2018. Presto: corpus noyau.

Robert D. Blumofe and Charles E. Leiserson. 1999. Scheduling multithreaded computations by work stealing. *J. ACM*, 46(5):720–748.

Bodleian Libraries. 2008–. Electronic enlightenment.

Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Melbourne, Australia. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Olivier Bonami and Sacha Beniamine. 2015. Implicative structure and joint predictiveness. In *Proceedings of the NetWordS Final Conference on Word Knowledge and Word Usage: Representations and Processes in the Mental Lexicon, Pisa, Italy, March 30 - April 1, 2015*, volume 1347 of *CEUR Workshop Proceedings*, pages 4–9. CEUR-WS.org.

Jan A. Botha, Emily Pitler, Ji Ma, Anton Bakalov, Alex Salcianu, David Weiss, Ryan McDonald, and Slav Petrov. 2017. Natural language processing with small feed-forward networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2879–2885, Copenhagen, Denmark. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Matthias Buch-Kromann. 2003. The danish dependency treebank and the dtag treebank tool. In *2nd Workshop on Treebanks and Linguistic Theories (TLT), Sweden*, pages 217–220.

Frederic Béchet and Eric Charton. 2010. Unsupervised knowledge acquisition for extracting named entities from speech. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5338–5341.

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. Clueweb09 data set.

Jean-Baptiste Camps, Simon Gabay, Paul Fièvre, Thibault Clérice, and Florian Cafiero. 2021. Corpus and Models for Lemmatisation and POS-tagging of Classical French Theatre. *Journal of Data Mining & Digital Humanities*, 2021.

Marie Candito and Benoît Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 138–141, Paris, France. Association for Computational Linguistics.

Marie Candito, Benoît Crabbé, and Pascal Denis. 2010. Statistical French dependency parsing: Treebank conversion and first results. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Marie Candito, Guy Perrier, Bruno Guillaume, Corentin Ribeyre, Karën Fort, Djamé Seddah, and Éric de la Clergerie. 2014. Deep syntax annotation of the sequoia French treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2298–2305, Reykjavik, Iceland. European Language Resources Association (ELRA).

Marie Candito and Djamé Seddah. 2012. Le corpus sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical (the sequoia corpus : Syntactic annotation and use for a parser lexical domain adaptation method) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN*, pages 321–334, Grenoble, France. ATALA/AFCP.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language ID in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Branden Chan, Timo Möller, Malte Pietsch, Tanay Soni, and Chin Man Yeung. 2019. German bert. `https://deepset.ai/german-bert`.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Thierry Chanier, Céline Poudat, Benoît Sagot, Georges Antoniadis, Ciara R. Wigham, Linda Hriba, Julien Longhi, and Djamé Seddah. 2014. The CoMeRe corpus for French: structuring and annotating heterogeneous CMC genres. *Journal for language technology and computational linguistics*, 29(2):1–30. Final version to Special Issue of JLCL (Journal of Language Technology and Computational Linguistics (JLCL, http://jlcl.org/): BUILDING AND ANNOTATING CORPORA OF COMPUTER-MEDIATED DISCOURSE: Issues and Challenges at the Interface of Corpus and Computational Linguistics (ed. by Michael Beißwenger, Nelleke Oostdijk, Angelika Storrer & Henk van den Heuvel).

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online. Association for Computational Linguistics.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 2635–2639. ISCA.

Avihay Chriqui and Inbal Yahav. 2021. HeBERT & HebEMO: a Hebrew BERT Model and a Tool for Polarity Analysis and Emotion Recognition. *arXiv e-prints*, page arXiv:2102.01909.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. CANINE: Pre-training an Efficient Tokenization-Free Encoder for Language Representation. *arXiv e-prints*, page arXiv:2103.06874.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Thibault Clérice. 2020. Pie extended, an extension for pie with pre-processing and post-processing.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

BNC Consortium et al. 2007. 520 million words, 1990-present. In *The British National Corpus, version 3 - BNC XML Edition*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Mark Davies. 2009. The 385+ million word corpus of contemporary american english (1990–2008+): Design, architecture, and linguistic insights. *International Journal of Corpus Linguistics*, 14(2):159–190.

Mark Davies. 2010. The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and Linguistic Computing*, 25(4):447–464.

DeLFT contributors. 2018. Delft. `https://github.com/kermitt2/delft`. [Online; accessed 16-August-2018].

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational*

Linguistics: EMNLP 2020, pages 3255–3265, Online. Association for Computational Linguistics.

Marie-Luce (dir.) Demonet. 1998–. Epistemon.

Spencer Desrochers, Chad Paradis, and Vincent M. Weaver. 2016. A validation of dram rapl power measurements. In *Proceedings of the Second International Symposium on Memory Systems*, MEMSYS '16, page 455–470, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. 2020. FQuAD: French question answering dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1193–1208, Online. Association for Computational Linguistics.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

Sascha Diwersy, Achille Falaise, Marie-Hélène Lay, and Gilles Souvay. 2017. Ressources et méthodes pour l'analyse diachronique. *Langages*, N° 206(2):21–44.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (*Volume 2: Short Papers*), pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Stefan Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. The birth of Romanian BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4324–4328, Online. Association for Computational Linguistics.

Yoann Dupont. 2017. Exploration de traits pour la reconnaissance d'entités nommées du français par apprentissage automatique (feature exploration for French named entity recognition with machine learning). In *Actes des 24ème Conférence sur le Traitement Automatique des Langues Naturelles. 19es REncontres jeunes Chercheurs en Informatique pour le TAL* (*RECITAL 2017*), pages 42–55, Orléans, France. ATALA.

Yoann Dupont and Isabelle Tellier. 2014. A named entity recognizer for French (un reconnaisseur d'entités nommées du français) [in French]. In *Proceedings of TALN 2014* (*Volume 3: System Demonstrations*), pages 40–41, Marseille, France. Association pour le Traitement Automatique des Langues.

Maud Ehrmann, Matteo Romanello, Alex Flückiger, and Simon Clematide. 2020. Extended overview of clef hipe 2020: Named entity processing on historical newspapers. In *CLEF 2020 Working Notes. Conference and Labs of the Evaluation Forum*, volume 2696 of *CEUR Workshop Proceedings. 2696*, page 38. CEUR-WS.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pages 5960–5969, Online. Association for Computational Linguistics.

Tomaž Erjavec. 2015. Reference corpus of historical slovene goo300k 1.2. Slovenian language resource repository CLARIN.SI.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Antoine Furetière. 1701. *Dictionnaire Universel, contenant généralement tous les mots françois tant vieux que modernes, & les termes des sciences et des arts*. Arnoud et Reinier Leers. Par Feu Messire Antoine Furetière,... 2e édition revue, corrigée et augmentée par M. Basnage de Bauval.

Simon Gabay. 2014. Pourquoi moderniser l'orthographe? principes d'ecdotique et littérature du XVIIe siècle. *Vox Romanica*, 73(1):27–42.

Simon Gabay, Alexandre Bartz, Philippe Gambette, and Alix Chagué. 2022. Freem max oa: A large corpus for early modern french - open access version.

Simon Gabay, Jean-Baptiste Camps, and Thibault Clérice. 2020. Manuel d'annotation linguistique pour le français moderne (XVIe -XVIIIe siècles).

Simon Gabay, Thibault Clérice, Jean-Baptiste Camps, Jean-Baptiste Tanguy, and Matthias Gille-Levenson. 2020-10. Standardizing linguistic data: method and tools for annotating (pre-orthographic) French. In *Proceedings of the 2nd International Digital Tools & Uses Congress (DTUC '20)*, Hammamet, Tunisia.

Simon Gabay, Pedro Ortiz Suarez, Alexandre Bartz, Alix Chagué, Rachel Bawden, Philippe Gambette, and Benoît Sagot. 2022. From FreEM to D'AlemBERT: a Large Corpus and a Language Model for Early Modern French. In *Proceedings of The 13th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association.

Sylvain Galliano, Edouard Geoffrois, Djamel Mostefa, Khalid Choukri, Jean-François Bonastre, and Guillaume Gravier. 2005. The ESTER phase II evaluation campaign for the rich transcription of French broadcast news. In *Proc. Interspeech 2005*, pages 1149–1152.

Sylvain Galliano, Guillaume Gravier, and Laura Chaubard. 2009. The ester 2 evaluation campaign for the rich transcription of French radio broadcasts. In *Proc. Interspeech 2009*, pages 2583–2586.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv e-prints*, page arXiv:2101.00027.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, III Daumé, Hal, and Kate Crawford. 2018. Datasheets for Datasets. *arXiv e-prints*, page arXiv:1803.09010.

Martin-Dietrich Gleßgen. 2003. L'élaboration philologique et l'étude lexicologique des plus anciens documents linguistiques de la France à l'aide de l'informatique. *Mémoires et documents de l'École des chartes*, 71:371–386.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *arXiv e-prints*, page arXiv:2106.03193.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Grobid contributors. 2008 — 2018. Grobid. `https://github.com/kermitt2/grobid`. [Online; accessed 16-August-2018].

Loïc Grobol and Benoit Crabbé. 2021. Analyse en dépendances du français avec des plongements contextualisés (French dependency parsing with contextualized embeddings). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, pages 106–114, Lille, France. ATALA.

Loïc Grobol, Mathilde Regnault, Pedro Ortiz Suarez, Laurent Romary, Benoît Sagot, and Benoît Crabbé. 2022. BERTrade: Using Contextual Embeddings to Parse Old French. In *Proceedings of The 13th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association.

Cyril Grouin, Sophie Rosset, Pierre Zweigenbaum, Karën Fort, Olivier Galibert, and Ludovic Quintard. 2011. Proposal for an extension of traditional named entities: From guidelines to evaluation, an overview. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 92–100, Portland, Oregon, USA. Association for Computational Linguistics.

Jean-Louis Guez de Balzac. 1624. *Lettres du sieur de Balzac*. T. Du Bray.

Gaël Guibon, Isabelle Tellier, Mathieu Constant, Sophie Prévost, and Kim Gerdes. 2014. Parsing Poorly Standardized Language Dependency on Old French. In *Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*, Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13), pages 51–61, Tübingen, Germany.

Céline Guillot, Serge Heiden, and Alexei Lavrentiev. 2018. Base de français médiéval : une base de référence de sources médiévales ouverte et libre au service de la communauté scientifique. *Diachroniques. Revue de Linguistique française diachronique*, 7:168–184.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2018. The Dataset Nutrition Label: A Framework To Drive Higher Data Quality Standards. *arXiv e-prints*, page arXiv:1805.03677.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv e-prints*, page arXiv:1508.01991.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Jean-Baptiste-Camps, LAKME-ENC, AliceCochet, LucenceIng, and Paulinelvq. 2019. Geste: Geste: un corpus de chansons de geste.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. FastText.zip: Compressing text classification models. *arXiv e-prints*, page arXiv:1612.03651.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the*

European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

David Kamholz, Jonathan Pool, and Susan Colowick. 2014. PanLex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3145–3150, Reykjavik, Iceland. European Language Resources Association (ELRA).

Vincentius Kevin, Birte Högden, Claudia Schwenger, Ali Şahan, Neelu Madan, Piush Aggarwal, Anusha Bangaru, Farid Muradov, and Ahmet Aker. 2018. Information nutrition labels: A plugin for online news evaluation. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 28–33, Brussels, Belgium. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Mohamed Khemakhem. 2020. *Standard-based Lexical Models for Automatically Structured Dictionaries*. Theses, Université de Paris.

Mohamed Khemakhem, Luca Foppiano, and Laurent Romary. 2017. Automatic extraction of tei structures in digitized lexical resources using conditional random fields. *electronic lexicography, eLex 2017*.

Mohamed Khemakhem, Axel Herold, and Laurent Romary. 2018. Enhancing usability for automatically structuring digitised dictionaries. *GLOBALEX workshop at LREC 2018*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. Greek-bert: The greeks visiting sesame street. In *11th Hellenic Conference on Artificial Intelligence*, SETN 2020, page 110–117, New York, NY, USA. Association for Computing Machinery.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Anthony Kroch, Ann Taylor, and Beatrice Santorini. 2000. The Penn-Helsinki Parsed Corpus of Middle English (PPCME2). *CD-ROM, second edition, release 4*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Pierre Kunstmann and Achim Stein. 2007. *Le nouveau corpus d'Amsterdam actes de l'atelier de Lauterbad, 23-26 février 2006*. Zeitschrift für französische Sprache und Literatur Neue Folge 34. F. Steiner, Stuttgart.

Anne Lacheret, Sylvain Kahane, Julie Beliao, Anne Dister, Kim Gerdes, Jean-Philippe Goldman, Nicolas Obin, Paola Pietrandrea, and Atanas Tchobanov. 2014. Rhapsodie: a prosodic-syntactic treebank for spoken French. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 295–301, Reykjavik, Iceland. European Language Resources Association (ELRA).

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513, Uppsala, Sweden. Association for Computational Linguistics.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020a. FlauBERT : des modèles de langue contextualisés pré-entraînés pour le français (FlauBERT : Unsupervised language model pre-training for French). In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles*, pages 268–278, Nancy, France. ATALA et AFCP.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020b. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

John Lee and Yin Hei Kong. 2012. A dependency treebank of classical Chinese poems. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 191–199, Montréal, Canada. Association for Computational Linguistics.

John Lee and Yin Hei Kong. 2014. A dependency treebank of Chinese Buddhist texts. *Digital Scholarship in the Humanities*, 31(1):140–151.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating Training Data Makes Language Models Better. *arXiv e-prints*, page arXiv:2107.06499.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramón Fermandez, Silvio Amir, Luís Marujo, and Tiago Luís. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, page arXiv:1907.11692.

Alexandra Luccioni and Joseph Viviano. 2021. What's in the box? an analysis of undesirable content in the Common Crawl corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 182–189, Online. Association for Computational Linguistics.

Enrique Manjavacas, Ákos Kádár, and Mike Kestemont. 2019. Improving lemmatization of non-standard languages with joint learning. 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)*, pages 1493–1503, Minneapolis, Minnesota. Association for Computational Linguistics.

Christiane Marchello-Nizia. 1979. *Histoire de la langue française aux XIVe et XVe siècles*. Collection Études. #0, Bordas, Paris.

Christiane Marchello-Nizia, Bernard Combettes, Sophie Prévost, and Tobias Scheer. 2020. *Grande Grammaire Historique du Français (GGHF)*. De Gruyter Mouton.

Elaine Marsh and Dennis Perzanowski. 1998. MUC-7 evaluation of IE technology: Overview of results. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Martin, Robert (dir.). 2020. *Dictionnaire du Moyen Français*. ATILF - CNRS & Université de Lorraine.

France Martineau. 2008. Un corpus pour l'analyse de la variation et du changement linguistique. *Corpus*, 7.

Mihai Masala, Stefan Ruseti, and Mihai Dascalu. 2020. RoBERT – a Romanian BERT model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6626–6637, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Philip May. 2019. German ELMo Model.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

Vincent Micheli, Martin d'Hoffschmidt, and François Fleuret. 2020. On the importance of pre-training data volume for compact language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7853–7858, Online. Association for Computational Linguistics.

Rada Mihalcea. 2007. Using Wikipedia for automatic word sense disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 196–203, Rochester, New York. Association for Computational Linguistics.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Aleksandra Miletic, Myriam Bras, Marianne Vergez-Couret, Louise Esher, Clamença Poujade, and Jean Sibille. 2020. Building a Universal Dependencies treebank for Occitan. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2932–2939, Marseille, France. European Language Resources Association.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Franco Moretti. 2013. *Distant reading*. Verso, London, New York.

Robert Morrissey and Mark Olsen. 1981–. American and french research on the treasury of the french language (artfl).

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

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Selinga, Lawrence Okegbemi, Laura Martinus, Kolawole Tajudeen, Kevin Degila, Kelechi Ogueji, Kathleen Siminyu, Julia Kreutzer, Jason Webster, Jamiil Toure Ali, Jade Abbott, Iroro Orife, Ignatius Ezeani, Idris Abdulkadir Dangana, Herman Kamper, Hady Elsahar, Goodness Duru, Ghollah Kioko, Murhabazi Espoir, Elan van Biljon, Daniel Whitenack, Christopher Onyefuluchi, Chris Chinenye Emezue, Bonaventure F. P. Dossou, Blessing Sibanda, Blessing Bassey, Ayodele Olabiyi, Arshath Ramkilowan, Alp Öktem, Adewale Akinfaderin, and Abdallah Bashir. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth David-

son, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Veronika Vincze,

Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.2. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Damien Nouvel, Jean-Yves Antoine, and Nathalie Friburger. 2014. Pattern mining for named entity recognition. In *Human Language Technology Challenges for Computer Science and Linguistics*, pages 226–237, Cham. Springer International Publishing.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [MASK]? Making Sense of Language-Specific BERT Models. *arXiv e-prints*, page arXiv:2003.02912.

Pedro Javier Ortiz Suárez, Yoann Dupont, Gaël Lejeune, and Tian Tian. 2020. Sinner@clef-hipe2020 : Sinful adaptation of sota models for named entity recognition in french and german. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Pedro Javier Ortiz Suárez, Yoann Dupont, Benjamin Muller, Laurent Romary, and Benoît Sagot. 2020a. Establishing a new state-of-the-art for French named entity recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4631–4638, Marseille, France. European Language Resources Association.

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

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. 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*, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, linguistic data consortium. *Technical report, Technical Report. Linguistic Data Consortium*.

Marco Passarotti. 2019. *The Project of the Index Thomisticus Treebank*, pages 299–320. De Gruyter Saur.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 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)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Addison Phillips and Mark Davis. 2005. Tags for Identifying Languages. Internet-Draft draft-phillips-langtags-10, Internet Engineering Task Force. Work in Progress.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Murielle Popa-Fabre, Pedro Javier Ortiz Suárez, Benoît Sagot, and Éric de la Clergerie. 2020. French contextualized word-embeddings with a sip of CaBeRnet: a new French balanced reference corpus. In *Proceedings of the 8th Workshop on Challenges in the Management of Large Corpora*, pages 15–23, Marseille, France. European Language Ressources Association.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Sophie Prévost and Achim Stein, editors. 2013. *Syntactic Reference Corpus of Medieval French (SRCMF)*, update version number edition. ENS de Lyon; Lattice, Paris; ILR University of Stuttgart, Lyon/Stuttgart.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Spencer Rarrick, Chris Quirk, and Will Lewis. 2011. MT detection in web-scraped parallel corpora. In *Proceedings of Machine Translation Summit XIII: Papers*, Xiamen, China.

Christian Raymond and Julien Fayolle. 2010. Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement. In *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles. Articles longs*, pages 191–200, Montréal, Canada. ATALA.

Pieter van Reenen, Margot van Mulken, and Evert Wattel. 2007. *Chartes de Champagne en français conservées aux Archives de l'Aube, 1270-1300*. Medievalia 61. Paradigme, Impr. Sagim-Canale, Orléans.

Reenen, Pieter van and Mulder, Maaike. 1998. Corpus middelnederlands.

Shruti Rijhwani, Daisy Rosenblum, Antonios Anastasopoulos, and Graham Neubig. 2021. Lexically aware semi-supervised learning for OCR post-correction. *Transactions of the Association for Computational Linguistics*, 9:1285–1302.

Vitor Rocio, Mário Amado Alves, J. Gabriel Lopes, Maria Francisca Xavier, and Graça Vicente. 2003. *Automated Creation of a Medieval Portuguese Partial Treebank*, pages 211–227. Springer Netherlands, Dordrecht.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Eiríkur Rögnvaldsson, Anton Karl Ingason, Einar Freyr Sigurðsson, and Joel Wallenberg. 2012. The Icelandic parsed historical corpus (IcePaHC). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1977–1984, Istanbul, Turkey. European Language Resources Association (ELRA).

Sophie Rosset, Cyril Grouin, and Pierre Zweigenbaum. 2011. Entités nommées structurées : guide d'annotation quaero. Technical report, LIMSI-CNRS. Autres.

Sophie Rosset, Gabriel Illouz, and Aurélien Max. 2005. Interaction et recherche d'information : le projet Ritel. *Traitement Automatique des Langues*, 46(3):155–179.

William Rothwell, Stewart Gregory, and David Trotter. 2005. *Anglo-Norman dictionary*, 2nd edition revised and enlarged edition. Publications of the modern humanities research Association vol. 17. Maney Publishing for the modern humanities research Association, London.

Benoît Sagot, Marion Richard, and Rosa Stern. 2012. Annotation référentielle du corpus arboré de Paris 7 en entités nommées (referential named entity annotation of the Paris 7 French TreeBank) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN*, pages 535–542, Grenoble, France. ATALA/AFCP.

Benoît Sagot and Rosa Stern. 2012. Aleda, a free large-scale entity database for French. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1273–1276, Istanbul, Turkey. European Language Resources Association (ELRA).

Manuela Sanguinetti and Cristina Bosco. 2015. *PartTUT: The Turin University Parallel Treebank*, pages 51–69. Springer International Publishing, Cham.

Helmut Schmid. 1999. *Improvements in Part-of-Speech Tagging with an Application to German*, pages 13–25. Springer Netherlands, Dordrecht.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green ai. *Commun. ACM*, 63(12):54–63.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Holger Schwenk and Jean-Luc Gauvain. 2005. Training neural network language models on very large corpora. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 201–208, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Stefan Schweter and Alan Akbik. 2020. FLERT: Document-Level Features for Named Entity Recognition. *arXiv e-prints*, page arXiv:2011.06993.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Shaked Greenfeld, and Reut Tsarfaty. 2021. AlephBERT:A Hebrew Large Pre-Trained Language Model to Start-off your Hebrew NLP Application With. *arXiv e-prints*, page arXiv:2104.04052.

Amit Seker, Amir More, and Reut Tsarfaty. 2018. Universal morpho-syntactic parsing and the contribution of lexica: Analyzing the ONLP lab submission to the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 208–215, Brussels, Belgium. Association for Computational Linguistics.

Satoshi Sekine and Chikashi Nobata. 2004. Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Linda J. Skitka, Kathleen L. Mosier, and Mark Burdick. 1999. Does automation bias decision-making? *International Journal of Human-Computer Studies*, 51(5):991–1006.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal Dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Achim Stein. 2014. Parsing heterogeneous corpora with a rich dependency grammar. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2879–2886, Reykjavik, Iceland. European Language Resources Association (ELRA).

Achim Stein. 2016. Old French dependency parsing: Results of two parsers analysed from a linguistic point of view. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 707–713, Portorož, Slovenia. European Language Resources Association (ELRA).

Rosa Stern. 2013. *Identification automatique d'entités pour l'enrichissement de contenus textuels*. Theses, Université Paris-Diderot - Paris VII.

Rosa Stern and Benoît Sagot. 2010. Resources for Named Entity Recognition and Resolution in News Wires. In *Entity 2010 Workshop at LREC 2010*, Valletta, Malta.

Rosa Stern, Benoît Sagot, and Frédéric Béchet. 2012. A joint named entity recognition and entity linking system. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pages 52–60, Avignon, France. Association for Computational Linguistics.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Milan Straka, Jana Straková, and Jan Hajič. 2019. Evaluating Contextualized Embeddings on 54 Languages in POS Tagging, Lemmatization and Dependency Parsing. *arXiv e-prints*, page arXiv:1908.07448.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Chenkai Sun, Abolfazl Asudeh, H. V. Jagadish, Bill Howe, and Julia Stoyanovich. 2019. Mithralabel: Flexible dataset nutritional labels for responsible data science. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2893–2896, New York, NY, USA. Association for Computing Machinery.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Wilson L. Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

TEI Consortium, eds. 2018. Tei p5: Guidelines for electronic text encoding and interchange. `http://www.tei-c.org/Guidelines/P5/`. [Version 3.4.0, 2018-07-23. Online; accessed 17-August-2018].

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. 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*, pages 142–147.

Elizabeth Closs Traugott and Susan Pintzuk. 2008. *Coding the York-Toronto-Helsinki Parsed Corpus of Old English Prose to investigate the syntaxpragmatics interface*, pages 61–80. De Gruyter Mouton.

Trieu H. Trinh and Quoc V. Le. 2018. A Simple Method for Commonsense Reasoning. *arXiv e-prints*, page arXiv:1806.02847.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv e-prints*, page arXiv:1908.08962.

Claire Hélène Vachon. 2010. *Le Changement linguistique au XVIe siècle: une étude basée sur des textes littéraires français*. ELiPhi, Éditions de linguistique et de philologie.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *arXiv e-prints*, page arXiv:1912.07076.

Denny Vrandečić and Markus Krötzsch. 2014-09-23. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020a. Neural machine translation with byte-level subwords. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9154–9160.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2643–2660, Online. Association for Computational Linguistics.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020b. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. In *Proceedings of the 1st Conference of*

the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 843–857, Suzhou, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv e-prints*, page arXiv:1910.03771.

David Wrisley. 2018. The open medieval french initiative (openmedfr).

Fei Wu and Daniel S. Weld. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. 2021. Nyströmformer: A nyström-based algorithm for approximating self-attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14138–14148.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950, Copenhagen, Denmark. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2021. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *arXiv e-prints*, page arXiv:2105.13626.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, Þórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali

Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, Lorena Martín-Rodríguez, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika

Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Shafi Sourov, Carolyn Spadine, Rachele Sprugnoli, Steinthór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021. Universal dependencies 2.9. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Þórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Luma Ateyah,

Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Ethan Chi, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Krister Lindén, Nikola Ljubešić, Olga Loginova, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mi-

titelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Arzucan Özgür, Balkız Öztürk Başaran, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Steinthór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, Hanzhi Zhu, and Anna Zhuravleva. 2020. Universal dependencies 2.7. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles

University.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

List of Figures

1.1 Oïl languages 8

4.1 A scheme of the `goclassy` pipeline. The red square represents the Compressed WET files stored on Amazon Web Services. The icons represent the gzip files stored locally, the represents one of the 50K WET files. The represents the filtered file and the represents a file of language tags, one tag per line in . The represents one of the 166 classified files. Each arrow represents an asynchronous non-blocking worker and dotted arrows represent a line filtering process. 36

6.1 Fraction of languages in each dataset below a given quality threshold (percent correct). 56
6.2 Percentage of sentences labeled as correct vs. log N sentences for all audited languages. 57

7.1 Record processing with metadata extraction. Headers are kept aside while sentences are identified and grouped into same-language bins. Headers are then cloned for each bin, and are sequentially stamped with an offset that is recorded for the whole operation, and written to disk into text and metadata files by language. 70
7.2 Comparison of language size (in bytes) between OSCAR 2018 and OSCAR 2021 (top/bottom 5 only). 72
7.3 Comparison of language percentage between OSCAR 2018 and OSCAR 2021 (top/bottom 5 only). 73

8.1 Corpus size distribution between OSCAR 21.09 and 22.01 82
8.2 Content size comparison of selected languages in OSCAR 22.01 versus OSCAR 21.09 83
8.3 Annotation count in selected low, mid and high resource languages (scales are adapted to corpus size) 86
8.4 Heat map of annotation distributions in selected low, mid and high resource languages. 87
8.5 Count of $(l1, l2)$ language tuples in the multilingual corpus. Languages tuples with less than 20,000 occurrences are not shown. . . . 88
8.6 Stacked distribution of annotated and non-annotated (clean) documents on a selection of the English corpus 89

10.1 Distribution of form and domain, gathered from documents metadata and manual annotation. 107
10.2 Distribution of the documents in the FreEM$_{max}$ corpus per year . . . 109
10.3 FreEM$_{max}$ compilation pipeline. All files are kept in their original format. Metadata is manually prepared in separate files in order to automatically transform and clean (in blue) all the available documents into XML TEI files following the same encoding. It allows us to distribute open data (in green) but also data distributed with restrictions regarding the modification of the original format (in orange). Non-open texts (in red) are not distributed. 110
10.4 Number of tokens per century. 111
10.5 Number of entities ($\log_{10}$ scale) per category. 114

E.1 Impact of number of pretraining steps on downstream performance for CamemBERT. 189

G.1 Number of entities by text on a logarithmic scale. 199
G.2 Entity types by text on a logarithmic scale. 200

List of Tables

1.1 Example of normalization taken from the *Lettres* of Guez de Balzac (1624). 10

2.1 Comparison of parallel and monolingual corpora extracted from web documents, including their downstream evaluation tasks. All parallel corpora are evaluated for machine translation (BLEU). TED-6: `da`, `cr`, `sl`, `sk`, `lt`, `et`; TED-45: 45-language subset of (Qi et al., 2018); WMT-5: `cs`, `de`, `fi`, `lv`, `ro`. POS/DEP-5: part-of-speech labeling and dependency parsing for `bg`, `ca`, `da`, `fi`, `id`. 17

3.1 Statistics on the treebanks used in POS tagging, dependency parsing, and NER (FTB). 28

4.1 Benchmarks are done using the UNIX `time` tool, are repeated 10 times each and are done for random samples of 10, 100 and 200 WET files. Only the classifying and filtering part are benchmarked. The table shows the minimum, maximum and mean time for the user, real and sys time over the 10 runs. Here "fastText" is used as short for the pipeline. 37

5.1 Size of Wikipedia corpora, measured in bytes, thousands of tokens, words and sentences. 42

5.2 Size of OSCAR 2019 subcorpora, measured in bytes, thousands of tokens, words and sentences. 42

5.3 Number of out-of-vocabulary words in random samples of 1M words for OSCAR 2019 and Wikipedia. 43

5.4 Size of treebanks, measured in thousands of tokens and sentences. . 45

5.5 Scores from UDPipe 2.0 (from Kondratyuk and Straka, 2019), the previous state-of-the-art models UDPipe 2.0+mBERT (Straka et al., 2019) and UDify (Kondratyuk and Straka, 2019), and our ELMo-enhanced UDPipe 2.0 models. Test scores are given for UPOS, UAS and LAS in all five languages. Best scores are shown in bold, second-best scores are underlined. 46

5.6 UPOS, UAS and LAS scores for the UDPipe 2.0 baseline reported by (Kondratyuk and Straka, 2019), plus the scores for checkpoints at 1, 3, 5 and 10 epochs for all the $\text{ELMo}_{\text{OSCAR}}$ and $\text{ELMo}_{\text{Wikipedia}}$. All scores are test scores. Best $\text{ELMo}_{\text{OSCAR}}$ scores are shown in bold while best $\text{ELMo}_{\text{Wikipedia}}$ scores are underlined. 48

6.1 Annotation codes for parallel data with sentence pair examples. The language code before each sentence indicates the language it is supposed to be in. 53

6.2 Averages of sentence-level annotations across datasets and selected languages. Macro-avg: Each language is weighted equally in the aggregation, regardless of its size. Micro-avg: Each label is weighted by the fraction of sentences for that language in the overall annotated corpus, i.e., the annotations for higher-represented languages are upweighted, and annotations for lower-represented languages are downweighted. The bottom rows contain the number of languages that have 0% labeled `C` etc. Note that these are not true expectations since the languages audited were not randomly sampled. 54

6.3 Rater evaluation for a subset of audits from **CCAligned** (aligned with English) measured by the accuracy (Acc-n) of annotations by non-proficient speaker against annotations by proficient speakers. . 58

6.4 Rater evaluation for a subset of audits from **OSCAR 2019** measured by the accuracy (Acc-n) of annotations by non-proficient speaker against annotations by proficient speakers. 58

6.5 Examples of "parallel" data where the translation has a different meaning than the source, but the form looks the same. (We added translations of the non-English side.) Such data may encourage hallucinations of fake "facts". 62

7.1 Comparison of approximate generation times depending on platform and number of shards. 68

7.2 Comparison of approximate generation times with and without metadata generation. 71

7.3 Comparison of the size of the Common Crawl dumps and their corresponding OSCAR sizes between the 2019 and the 21.09 versions. Compressed (Common Crawl) sources are from November 2018 and February 2021 dumps. Total is Textual + Metadata without deduplication. 72

7.4 Comparison of occurrences of news-related terms between OSCAR and our corpus in a sample of 100 Common Crawl shards. For the Burmese language, we use the whole 2018 and 2021 corpus since it is a low resource language. Terms are translated to the target language. 74

8.1 Comparison of occurrences of news-related terms between OSCAR and our corpus in a sample of 100 Common Crawl shards. *: For the Burmese language, we use the whole 21.09 and 22.01 corpus since it has less than 100 sentences in both subcorpora. Terms are translated in the corpus language. 85

9.1 Comparison of number of unique forms in the different genres represented by CaBeRnet partition. TTR: Type-Token Ratio. Lemmatization and tokenization was performed as described in §9.1.3. 97
9.2 Lexical statistics of French CBT, performed as described in §9.1.3 . . . 99
9.3 Comparing the corpora under study. 100
9.4 Lexical statistics on morphological richness over randomly selected samples of 3 million words from each corpus. nb : number 101

10.1 Data collection 106
10.2 Breakdown of the FrEEM$_{max}$ corpus by text origin. 109
10.3 Types (in gray) and subtypes taken from the *Quaero* typology. 112
10.4 NERC Fine-Grained annotation avec EL 112

11.1 **POS** and **dependency parsing** scores on 4 French treebanks, reported on test sets assuming gold tokenization and segmentation (best model selected on validation out of 4). Best scores in bold, second best underlined. 122
11.2 **NER** scores on the FTB (best model selected on validation out of 4). Best scores in bold, second best underlined. 123
11.3 **NLI** accuracy on the French XNLI test set (best model selected on validation out of 10). Best scores in bold, second best underlined. . . 123
11.4 Statistics on the pre-training datasets used. 125
11.5 Results on the four tasks using language models pre-trained on data sets of varying homogeneity and size, reported on validation sets (average of 4 runs for POS tagging, parsing and NER, average of 10 runs for NLI). 126

12.1 Results on the test set for the best development set scores. 133
12.2 Results on the test set for the best development set scores. 135
12.3 Final POS and dependency parsing scores on 4 French treebanks (French GSD, Spoken, Sequoia and ParTUT), reported on test sets (4 averaged runs) assuming gold tokenisation. Best scores in bold, second to best underlined, state-of-the-art results in italics. 136
12.4 NER Results on French Treebank (FTB): **best scores**, second to best. . 138

13.1 Statistics on the training and development data in French and German 142
13.2 Example extracted from the French training dataset 142

13.3 Strict results for our systems compared to the winning system (micro measures) . . . 145
13.4 Comparison between segments and sentences on French dev dataset (run 1), strict scenario . . . 146
13.5 Results obtained on the test set (strict metric) with only the train set (not to dev) and with train+dev sets (to dev) with our best system (run 2) . . . 146

14.1 Results on SRCMF dev — no additional data. . . . 151
14.2 Results on SRCMF dev — monolingual models. . . . 152
14.3 Results on SRCMF dev — Performances of different model sizes when training from scratch . . . 153
14.4 Results on SRCMF dev — using raw data. . . . 153
14.5 Results on SRCMF test . . . 154

15.1 Comparison between D'AlemBERT, CamemBERT and Pie Extended performance on the test set, out-of-domian data of FreEM$_{LPM}$. Best scores in **bold** sencond best in *italics*. . . . 157
15.2 Comparison between D'AlemBERT, CamemBERT and an LSTM-CRF-based model performance on the test set of FreEM$_{NER}$. Best scores in **bold** sencond best in *italics*. . . . 158

A.1 Size of the OSCAR corpus by language measured in bytes and number of words. Standard UNIX human-readable notation is used for the size in byte. We define "words" as spaced separated tokens, which gives a good estimate of the size of each corpus for languages using Latin or Cyrillic alphabets, but might give a misleading size for other languages such as Chinese or Japanese. . . . 169

B.1 Average power draw (Watts), training times (in both hours and days), mean power consumption (KWh) and CO_2 emissions (kg) for each ELMo model trained. . . . 172
B.2 Number of training steps for each checkpoint, for the ELMo$_{\text{Wikipedia}}$ and ELMo$_{\text{OSCAR}}$ of each language. . . . 173

C.1 Situations where two language codes are represented, but one is a superset of another by the ISO standard, leading to unclarity about the data in the supercode dataset. *The `als` dataset is actually in `gsw`. 175
C.2 There are 48 languages in the JW300 corpus with language codes that correspond to sign languages, but in reality are unrelated high-resource languages (usually the most spoken language in the country of origin of the sign language). This table shows the actual language of the data corresponding to each sign language code. . . . 176

C.3 Language code issues in the JW300 datasets for 22 language varieties not covered by Tables C.1 and C.2. Private use extensions are given as they appear in `jw.org`, and specified as '?' if they are absent from `jw.org`. 177
C.4 Miscellaneous errors in language codes. 178
C.5 Audit results for a sample of 100 sentences from **OSCAR** for each language, compared to the number of sentences available in the dataset. If fewer than 100 sentences were available, all sentences were audited Language codes are as originally published. Length is measured in number of characters. Languages with less than 20% correct sentences are boldfaced. 181

D.1 Size of the OSCAR 22.01 corpus by language measured in bytes and number of words. Standard UNIX human-readable notation is used for the size in byte. We define "words" as spaced separated tokens, which gives a good estimate of the size of each corpus for languages using Latin or Cyrillic alphabets, but might give a misleading size for other languages such as Chinese or Japanese. 185

E.1 Performance reported on **Test sets** for all trained models (**average** over multiple fine-tuning seeds). 187
E.2 Comparing scores on the **Validation sets** of different design choices. POS tagging and parsing datasets are averaged. (average over multiple fine-tuning seeds). 188

F.1 Average power draw, number of models trained, training times in hours, mean power consumption including power usage effectiveness (PUE), and CO_2 emissions; for each setting. 193

G.1 Average power draw, number of models trained, training times in hours, mean power consumption including power usage effectiveness (PUE), and CO^2 emissions; for each setting. 195
G.2 Results of the BiLSTM-CRF model on the test set of FrEM$_{NER}$ by entity type. 197
G.3 Results of CamemBERT on the test set of FrEM$_{NER}$ by entity type. . 197
G.4 Results of D'AlemBERT model on the test set of FrEM$_{NER}$ by entity type. 198

License

This document is available under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0) (https://creativecommons.org/licenses/by/4.0/deed.en)

Copyright © 2022, Pedro Ortiz Suarez <pedro@portizsu.eu>



Une approche basée sur les données pour le traitement automatique du langage naturel en français contemporain et historique

Depuis plusieurs années, les approches neuronales ont régulièrement amélioré l'état de l'art du traitement automatique des langues (TAL) sur une grande variété de tâches. L'un des principaux facteurs ayant permis ces progrès continus est l'utilisation de techniques d'apprentissage par transfert. Ces méthodes consistent à partir d'un modèle pré-entraîné et à le réutiliser, avec peu ou pas d'entraînement supplémentaire, pour traiter d'autres tâches. Même si ces modèles présentent des avantages évidents, leur principal inconvénient est la quantité de données nécessaire pour les pré-entraîner. Ainsi, le manque de données disponibles à grande échelle a freiné le développement de tels modèles pour le français contemporain et a fortiori pour ses états de langue plus anciens.

Cette thèse met l'accent sur le développement de corpus pour le pré-entraînement de telles architectures. Cette approche s'avère extrêmement efficace car nous sommes en mesure d'améliorer l'état de l'art pour un large éventail de tâches de TAL pour le français contemporain et historique, ainsi que pour six autres langues contemporaines. De plus, nous montrons que ces modèles sont extrêmement sensibles à la qualité, à l'hétérogénéité et à l'équilibre des données de pré-entraînement et montrons que ces trois caractéristiques sont de meilleurs prédicteurs de la performance des modèles que la taille des données de pré-entraînement. Nous montrons également que l'importance de la taille des données de pré-entraînement a été surestimée en démontrant à plusieurs reprises que l'on peut pré-entraîner de tels modèles avec des corpus de taille assez modeste.

Mots-clés : modèle de langue, corpus de pré-entraînement, traitement automatique des langues, français contemporain, français historique, apprentissage par transfert.

A Data-driven Approach to Natural Language Processing for Contemporary and Historical French

In recent years, neural methods for Natural Language Processing (NLP) have consistently and repeatedly improved the state of the art in a wide variety of NLP tasks. One of the main contributing reasons for this steady improvement is the increased use of transfer learning techniques. These methods consist in taking a pre-trained model and reusing it, with little to no further training, to solve other tasks. Even though these models have clear advantages, their main drawback is the amount of data that is needed to pre-train them. The lack of availability of large-scale data previously hindered the development of such models for contemporary French, and even more so for its historical states.

In this thesis, we focus on developing corpora for the pre-training of these transfer learning architectures. This approach proves to be extremely effective, as we are able to establish a new state of the art for a wide range of tasks in NLP for contemporary, medieval and early modern French as well as for six other contemporary languages. Furthermore, we are able to determine, not only that these models are extremely sensitive to pre-training data quality, heterogeneity and balance, but we also show that these three features are better predictors of the pre-trained models' performance in downstream tasks than the pre-training data size itself. In fact, we determine that the importance of the pre-training dataset size was largely overestimated, as we are able to repeatedly show that such models can be pre-trained with corpora of a modest size.

Keywords: language model, pre-training corpora, natural language processing, contemporary french, historical french, transfer learning.

Sorbonne Université - École doctorale EDITE (ED130)
Campus Pierre et Marie Curie, Tour 24-25 bureau 210
4, place Jussieu, 75252 Paris Cedex 05, France