ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

School of Information Sciences and Technology
Department of Informatics
Athens, Greece

MSc Thesis
in
Data Science

Large Pre-trained Language Model for Contractual and Regulatory text

Sotirios Legkas

Supervisors: Dr. Prodromos Malakasiotis
Department of Informatics
Athens University of Economics and Business

Dr. Ilias Chalkidis
Cognitiv+ Document Intelligence, Athens
Department of Computer Science
University of Copenhagen, Denmark

October 2022

Sotirios Legkas

Large Pre-trained Language Model for Contractual and Regulatory text

October 2022

Supervisor: Dr. Prodromos Malakasiotis

Athens University of Economics and Business

School of Information Sciences and Technology

Department of Informatics

Athens, Greece

Abstract

The excessive amount and length of legal documents create difficulties in analysis by humans. For that reason, many start-ups operate on the artificial intelligence (AI) field providing text analytics through deep learning techniques that extract useful insights from documents. Start-ups have to adapt their technology accordingly to the trends of billion-parameter-sized Language Models (LMs). However, challenges arise due to high computational resources and economic consequences for the development and deployment for such large models. This work provides important insights on the aforementioned problem, as it follows the steps of the R&D group of a modern legal-tech start-up. Following Chalkidis et al., 2020, who showed that domain-specific models in legal corpora perform better in several legal NLP tasks, we pre-train from scratch multiple variants of a domain-specific multi-lingual LM based on RoBERTa architecture. The corpus that was used is consisted by contractual and regulatory legal text in ten different languages. The goal is to re-use the pre-trained models as the backbone of any model in the future to train multiple classifiers of interest with limited training data compared to those needed to have equal performance with a non pre-trained model. Finally, the performance of the models is benchmarked across 5 down-stream legal NLP tasks, which comprise both publicly available and private datasets covering both English and multi-lingual datasets and several task types (document/sentence classification, natural language inference, and entity extraction). The results suggest that larger domain-specific models outperform smaller ones and that our domain-specific C-XLM models outperform their corresponding generic XLM-R models, even though they use smaller vocabulary and are pre-trained for fewer steps. Lastly, smaller domain-specific models achieve competitive results against larger generic models. Concluding, pre-training a reasonably large LM for contract and regulatory related tasks obtain top-notch classification results compared to smaller and less domain-specific models.

Περίληψη

Η ανάλυση πολλαπλών και ιδιαίτερα εκτενών νομικών εγγράφων απαιτεί πολύωρη ενασχόληση αυτών που τα χρησιμοποιούν. Για το λόγο αυτό, πολλές νεοφυείς επιχειρήσεις δραστηριοποιούνται στον κλάδο της τεχνητής νοημοσύνης (ΤΝ) παρέχοντας ανάλυση κειμένων μέσω τεχνικών βαθιάς μάθησης, που εξάγουν χρήσιμες πληροφορίες από τα έγγραφα. Οι νεοφυείς επιχειρήσεις πρέπει να προσαρμόσουν την τεχνολογία τους ανάλογα με τις τάσεις των γλωσσικών μοντέλων (ΓΜ) δισεκατομμυρίων παραμέτρων. Ωστόσο, η ανάπτυξη μεγάλων μοντέλων είναι δύσκολη εξαιτίας των υψηλών απαιτήσεων σε υπολογιστικούς πόρους και των οικονομικών συνεπειών από την ανάπτυξη και την ενσωμάτωση τους σε μία εφορμογή. Η παρούσα μελέτη παρέχει σημαντικές πληροφορίες σχετικά με το προαναφερθέν πρόβλημα, καθώς ακολουθεί τα βήματα του τμήματος έρευνας και ανάπτυξης μιας νεοφυούς επιχείρησης, εξειδικευμένης στην εφαρμογή τεχνολογίας ΤΝ με τεχνικές βαθιάς μάθησης σε νομικά κείμενα. Ακολουθώντας την μελέτη του Χαλκίδη κ.ά., 2020, που έδειξε ότι τα προ-εκπαιδευμένα μοντέλα στον συγκεκριμένο τομέα των νομικών κειμένων αποδίδουν καλύτερα σε διάφορες νομικές διεργασίες επεξεργασίας φυσικής γλώσσας (ΕΦΓ), προ-εκπαιδεύουμε από την αρχή πολλαπλές παραλλαγές ενός πολυγλωσσικού ΓΜ σε νομικά κείμενα με βάση την αρχιτεκτονική του μοντέλου RoBERTa. Τα κείμενα που χρησιμοποιήθηκαν αποτελούνται από συμβατικά και κανονιστικά νομικά κείμενα σε δέκα διαφορετικές γλώσσες. Στόχος είναι η εκ νέου χρήση των προ-εκπαιδευμένων μοντέλων ως ραχοκοκαλιά οποιουδήποτε μοντέλου στο μέλλον για την εκπαίδευση πολλαπλών ταξινομητών που μας ενδιαφέρουν, με περιορισμένα δεδομένα εκπαίδευσης σε σύγκριση με εκείνα που απαιτούνται για να έχουν ίση απόδοση με ένα μη προ-εκπαιδευμένο μοντέλο. Τέλος, η απόδοση των μοντέλων έχει ως σημεία αναφοράς 5 νομικές διεργασίες ΕΦΓ, οι οποίες αποτελούνται από δημόσια και ιδιωτικά σύνολα δεδομένων. Τα σύνολα δεδομένων απαρτίζονται απο 3 αγγλικά και 2 πολυγλωσσικά σύνολα διαφόρων τύπων διεργασιών (ταξινόμηση εγγράφων/προτάσεων, συμπερασματολογία φυσικής γλώσσας και εξαγωγή οντοτήτων). Τα αποτελέσματα έδειξαν ότι τα μεγαλύτερα μοντέλα που εκπαιδεύτηκαν σε δεδομένα συγκεκριμένου τομέα κειμένων αποδίδουν καλύτερα από τα μικρότερα μοντελα και ότι τα μοντέλα C-XLM μας ξεπερνούν σε απόδοση τα αντίστοιχα μοντέλα XLM-R που έχουν εκπαιδευτεί σε δεδομένα κειμένων γενικής φύσεως, παρόλο που χρησιμοποιούν μικρότερο λεξιλόγιο και έχουν προ-εκπαιδευτεί για λιγότερα βήματα. Επιπλέον, τα μικρότερα μοντέλα που εκ-

παιδεύτηκαν στον τομέα νομικών εγγράφων επιτυγχάνουν ανταγωνιστικά αποτελέσματα έναντι μεγαλύτερων γενικών μοντέλων. Εν κατακλείδι, η προ-εκπαίδευση ενός αρκετά μεγάλου ΓΜ για διεργασίες συσχετιζόμενες με συμβάσεις και κανονιστικά νομικά κείμενα επιτυγχάνει κορυφαία αποτελέσματα ταξινόμησης σε σύγκριση με μικρότερα και μη εξειδικευμένα στον τομέα μοντέλα.

Acknowledgements

First of all, I would like to sincerely thank my supervisor Prodromos Malakasiotis, who offered me constant guidance and advice throughout the period of this work. Also, I would like to thank Ilias Chalkidis for all the valuable help and for the great cooperation on a regular basis. I would especially like to thank them for sharing their expertise and taught me so many things regarding NLP. Furthermore, I would like to thank them for introducing me in the NLP research field and having my first publication co-authored with them. Next, I would like to thank Cognitiv+ and specifically Vasilis Tsolis for providing me all the necessary tools and support needed for the completion of this project. Moreover, I would like to thank all my fellow students without any exception for their mutual help throughout this year, but more importantly for their friendship and their support when most needed. Last but not least, I would like to wholeheartedly express my gratitude and thank most my family for their endless support and encouragement during all those years of studying.

Contents

Abstract vi

Acknowledgements vii

1 Introduction 1

1.1 Motivation and Problem Statement 1

1.2 Thesis Structure 3

2 Background and Related Work 5

2.1 Transformers 5

2.2 Transformer-Based Language Models 8

2.3 Domain-Specific Language Models 12

2.4 Transformer-Based Multi-lingual Models 12

2.5 Evaluation Measures 14

3 Model Specifications 17

4 Pre-Training 19

4.1 Training Corpora 19

4.2 Custom Vocabulary 20

4.3 Masked Language Modeling (MLM) 21

4.4 MLM Results 21

5 Fine-Tuning 25

5.1 Benchmark - Tasks and Datasets 25

5.2 Experimental Set Up 28

5.3 Fine-tuning Results 28

6 Conclusions 35

Bibliography 37

List of Acronyms 43

List of Figures 45

List of Tables 46

Introduction

1

1.1 Motivation and Problem Statement

Many companies and start-ups operate in artificial intelligence (AI) industries providing text analytics that help extracting useful insights from documents. The large variety of documents (legal, scientific, biomedical etc), along with the excessive amount and length of these documents create difficulties in analysis by humans. For that reason, big companies and start-ups are trying to provide new intelligent document services. The public access to many legal documents and the continuous innovative field of deep learning contribute more and more AI services with fascinating performance that improves year by year.

Natural Language Processing (NLP) is a continuously growing field, which had tremendous evolution the past few years with the inclusion of deep learning technologies. Specifically, legal text processing is an extremely important aspect of NLP that can provide useful insights in the legal field. Researchers applied deep learning techniques in legal documents aiming at tasks such as legal topic classification, detection of unfair clauses in Terms of Service Agreements, legal judgment prediction and analysis, legal question answering, contract based Natural Language Inference in Non-Disclosure Agreements, obligation extraction from contracts and contract element extraction. The key points that distinguishes legal documents (legislation, court cases, legal agreements) from other types are the following [35]:

- Great length
- Long sentences
- Specific structure
- Specialized vocabulary (legal terminology, use of French and Latin)
- Syntax
- In-domain meaning of words
- Use of pronominal adverbs (e.g. herein, whereby, wherefore etc.)
- Unusual word order
- Use of bullet points
- Writing idioms

For these reasons, the legal NLP literature is also flourishing with the release of many new resources; including large legal corpora [32], benchmark datasets [13, 17, 39], and pre-trained legal-oriented language models [15, 69].

Also, transformer-based Languages Models (LMs) [55, 22, 46] have stormed the NLP community achieving state-of-the-art results in various NLP tasks. A historical paradigm swift occurred in the post 2018 era, where instead of training shallow randomly initialized models for several epochs (iterations), deep pre-trained models are fine-tuned for a few epochs. These models can be potentially fine-tuned to resolve tasks of interest (e.g., contract element extraction, obligation extraction, etc.). In recent years, humongous billion-parameter-sized models have been developed [9, 56, 34] showcasing impressive few-shot capabilities. Lastly, recent research results show that domain-specific models (e.g., pre-trained in legal corpora [15, 69], biomedical corpora [43], clinical corpora [3], scientific corpora [7]) perform better in domain-specific down-stream tasks.

Moreover, multi-lingual models [21, 20] have been developed, which are capable of "learning" and adapting to several languages. Concerning down-stream tasks, these models have also the potential to adapt from one (source) language to others (targets) with limited or no training data in the target languages and without an absurd performance drop. This is evident from the exceptional results reported by multi-lingual LMs in several multi-lingual benchmarks, and especially in zero-shot cross-lingual transfer [22, 21, 41, 20, 26].

Summarizing, in the era of billion-parameter-sized Language Models (LMs), start-ups have to follow trends and adapt their technology accordingly. Nonetheless, there are open challenges since the development and deployment of large models come with a need for large compute resources and has economic consequences. In this work, we follow the steps of the R&D group of a modern legal-tech start-up and present important insights on model development and deployment. We start from ground zero by pre-training multiple domain-specific multi-lingual LMs, which fit better contractual and regulatory text compared to available alternatives (XLM-R [20]). Given the above observations, we believe that pre-training a reasonably large language model for contract and regulatory related tasks has a great potential for top-notch classification results compared to re-using smaller and less domain-specific models. The goal is to pre-train a large multi-lingual model and re-use it as the backbone of any model in the future to train multiple classifiers of interest with limited training data compared to those needed to have equal performance with a non pre-trained model. We present benchmark results of such models in a half-public half-private legal benchmark comprising 5 down-stream tasks showing the impact of larger model size.

Despite the impressive progress, the efficacy of differently-sized language models on legal NLP tasks, and the importance of legal domain legal specificity is still understudied. In this work, we aim to shed light across all these directions following model development across two incremental steps: (a) *model pre-training* on large legal corpora, and (b) *model fine-*

tuning on down-stream tasks. To do so, we initially develop 4 multi-lingual legal-oriented language models (C-XLMs). We benchmark their performance across 5 down-stream legal NLP tasks, which comprise both publicly available and private datasets covering both English and multi-lingual datasets and several task types (document/sentence classification, natural language inference, and entity extraction).

Following Chalkidis et al. [15], who created from scratch different size variations of LEGAL-BERT, our work aims to provide guidelines to legal-tech practitioners on model development (pre-training, fine-tuning) bearing both performance and efficiency into consideration. The reason is that the full capacity of larger and computationally more expensive models may be unnecessary in specialized domains, where syntax may be more standardized, the range of topics discussed may be narrower, terms may have fewer senses etc. In general, we study the impact of larger vs. smaller models, domain-specific vs. generic models, and find that larger domain-specific models perform better.

Our main research questions could be summed up as:

- Do multi-lingual, legal domain-specific, RoBERTa [46] based models outperform the multi-lingual XLM-RoBERTa [20] models in legal NLP tasks (document/sentence classification, natural language inference, and entity extraction)?
- Do our larger domain-specific language models significantly outperform smaller ones?
- Do our smaller domain-specific language models perform better/similar to XLM-R models?

1.2 Thesis Structure

The rest of the thesis is structured as :

Chapter 2 presents the background needed for the understanding of this thesis, along with the related work over the years, regarding transformer-based language models, domain-specific language models and transformer-based multi-lingual models.

Chapter 3 provides information about the specifications of the different sized transformer-based models that we pre-trained from scratch in multi-lingual legal domain-specific corpus.

Chapter 4 reports details considering the training corpora, the custom vocabulary and the training of the Masked Language Models, followed by the obtained results.

Chapter 5 discusses the evaluation benchmarks, analyzing details regarding the datasets of each task and the task itself. Following, the experimental set up and the corresponding results are being presented for each down-stream task.

Chapter 6 concludes, providing a summary of the findings, along with highlights, limitations and suggests ideas for future work

2 Background and Related Work

This section provides details on the necessary background needed for the rest of this thesis. The transformer architecture, transformer-based models (BERT, RoBERTa) and all the related concepts will be analyzed, along with the evaluation measures that will be used during the thesis. In addition, work related to pre-trained, multi-lingual and domain-specific models will be presented.

2.1 Transformers

Following, the success of RNNs (Recurrent Neural Networks) in NLP that captures the sequences from left to right and vice versa, Vaswani et al. [63] introduced a great innovation in the NLP field called transformer. Transformer is an attention-based neural network, consisted of at least an **encoder** (left part of Figure 2.1) and a **decoder** (right part of Figure 2.1). More than one encoder and decoder can create the encoder stack and decoder stack, as it can be seen in Figure 2.2. It was initially used mostly for sequence-to-sequence problems like machine translation. The main aspect of the newly introduced transformer is that it uses an "attention" mechanism to look the entire sequence at once, without the use of recurrence and provide an attention score in each element of a sequence based on its importance. The architecture of the transformer is depicted in Figure 2.1 and the related components will be analyzed below.

Vaswani et al. [63] used Byte-Pair Encoding (BPE) for tokenization, which was first introduced by Sennrich et al. [58]. BPE is a process which pre-tokenizes text into words and afterwards creates a set of unique words along with their frequency. Then, it turns the unique words into a set of symbols, followed by merging a pair of symbols and creating a new symbol-pair based on the frequency of this pair. This process is continued until the number of symbols reaches the vocabulary size (the number of merges and the vocabulary size are hyper-parameters).

Afterwards, the tokens are converted into **embedding vectors**, which are the representations of tokens with a set of weights from a normal distribution N(0,1). The dimension of these embeddings is $R^{|V| * 512}$, where |V| is the vocabulary size and 512 is the dimensionality of input and output as defined in the paper.

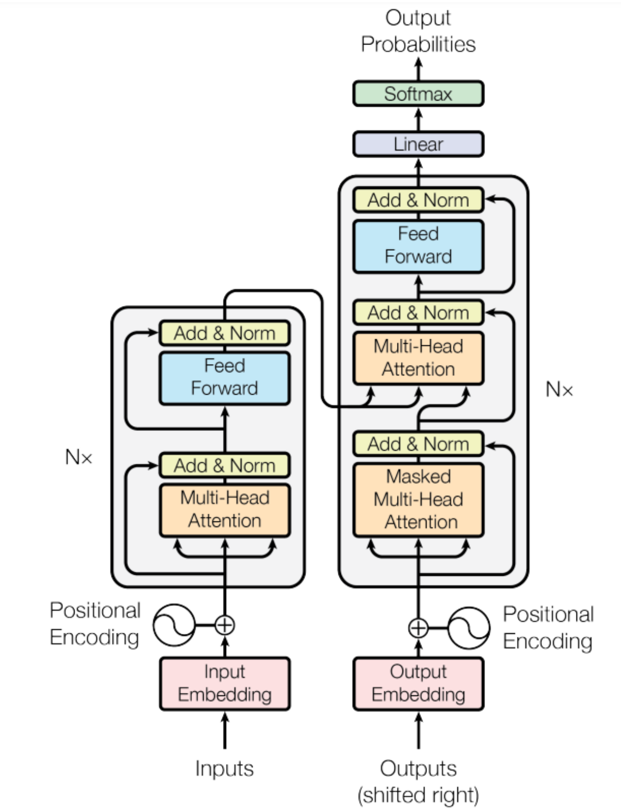


Fig. 2.1: The transformer-model architecture. The encoder is the left part of the transformer and the decoder is the right part. Source: Vaswani et al. [63]

Since the processing of the sequence is happening all at once, the position of the token in the sequence is unknown. Hence, the transformer adds a **positional encoding vector** to each token embedding and the resulting embedding is the input of the encoder. The construction of the input of the encoder is depicted in Figure 2.3

The first layer of the encoder is the **self-attention layer**, which is the key mechanism in the transformers. This mechanism features a sequential process that detects related tokens in the same sequence, regardless of the position of each token. Instead of looking left to right or vice versa, it uses this mechanism to look the whole sequence at once and assign an attention score to each token of the sequence, according to its importance. The attention score is calculated through matrix multiplication operations for each token in the sequence. The attention mechanism is also called an attention head, and there can be more than one such mechanisms, constructing a Multi-head attention layer. Each different head captures different structures and meanings in a sequence, that may start with simpler and easier meanings to more complex and specific ones. The benefits of the Multi-head attention are that operations can be executed in parallel, the reduction of computational complexity and the higher length of the sequence that can be used compared to other methods.

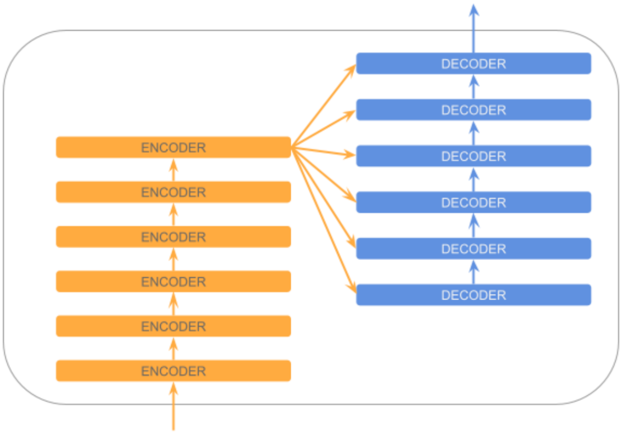


Fig. 2.2: Transformer model with Encoder Stack-Decoder Stack. Source: `https://www.baeldung.com/cs/transformer-text-embeddings`

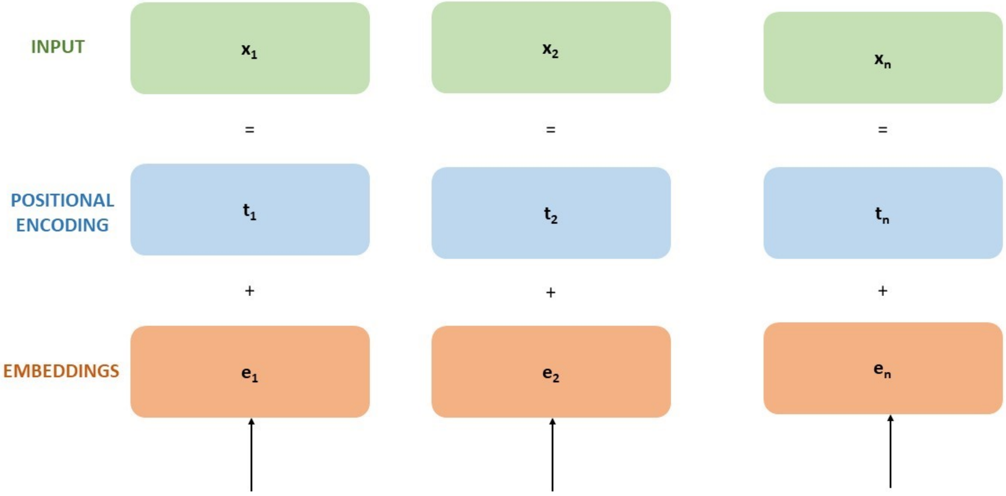


Fig. 2.3: Construction of the input of the encoder from embedding vectors and positional encoding vectors. Source: `https://www.baeldung.com/cs/transformer-text-embeddings`

The next step called **Add & Norm**. Add called in Figure 2.1 refers to residual connection [30], that adds the input of each layer to the output, and is employed around both the multi-head self-attention mechanism sub-layer and the position-wise fully connected feed-forward network sub-layer (Figure 2.1). Norm refers to layer normalization [4], which normalize the output of the residual connection in order to follow a normal distribution. The last step is to pass the output of Add & Norm through a feed-forward network and use another Add & Norm layer to produce the final output of the encoder.

Each encoder provides an output which is used as an input on the next encoder. The last encoder provides the output of the whole encoder stack (Figure 2.2). Each decoder takes as input a previously generated sequence and the encoder's output. A decoder contains the same elements as the encoder but with the addition of an encoder-decoder attention layer, which provides insights about which tokens of the input sequence are more relevant to the output token. The decoder stack output produces a vector of floats that is processed through a **linear layer** and is projected into a vector with the same size as the vocabulary. Finally, **softmax** is applied to this vector converting the scores into probabilities, meaning that the token with the highest probability is the next token that should be picked.

2.2 Transformer-Based Language Models

BERT

Devlin et al. [22] from Google, created a model called BERT, which stands for Bidirectional Encoder Representation from transformers. The model was deployed by Google's search engine in 2019. It is a transformer-based language model, pre-trained on deep bidirectional representations from unlabeled data, taking into consideration both left and right context in all layers. The corpus that was used in the pre-training included BookCorpus with 800M words and English Wikipedia with 2.5B words.

This model can be fine-tuned by adding one extra output layer for implementation of NLP tasks like text classification, named entity recognition, question answering, language modeling, summarization, translation, multiple choice, and text generation. Figure 2.4 depicts the illustration of pre-training and fine-tuning of BERT, while Figure 2.5 depicts the fine-tuning of BERT on several NLP tasks in more detail.

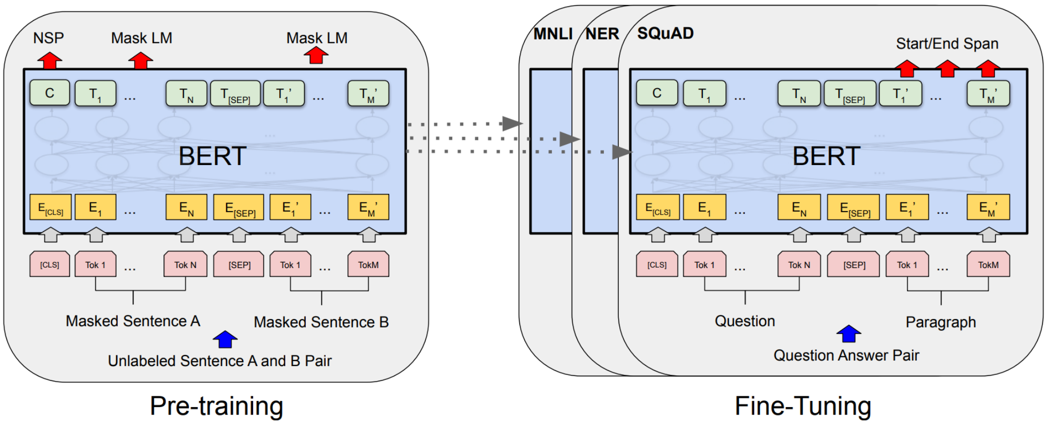
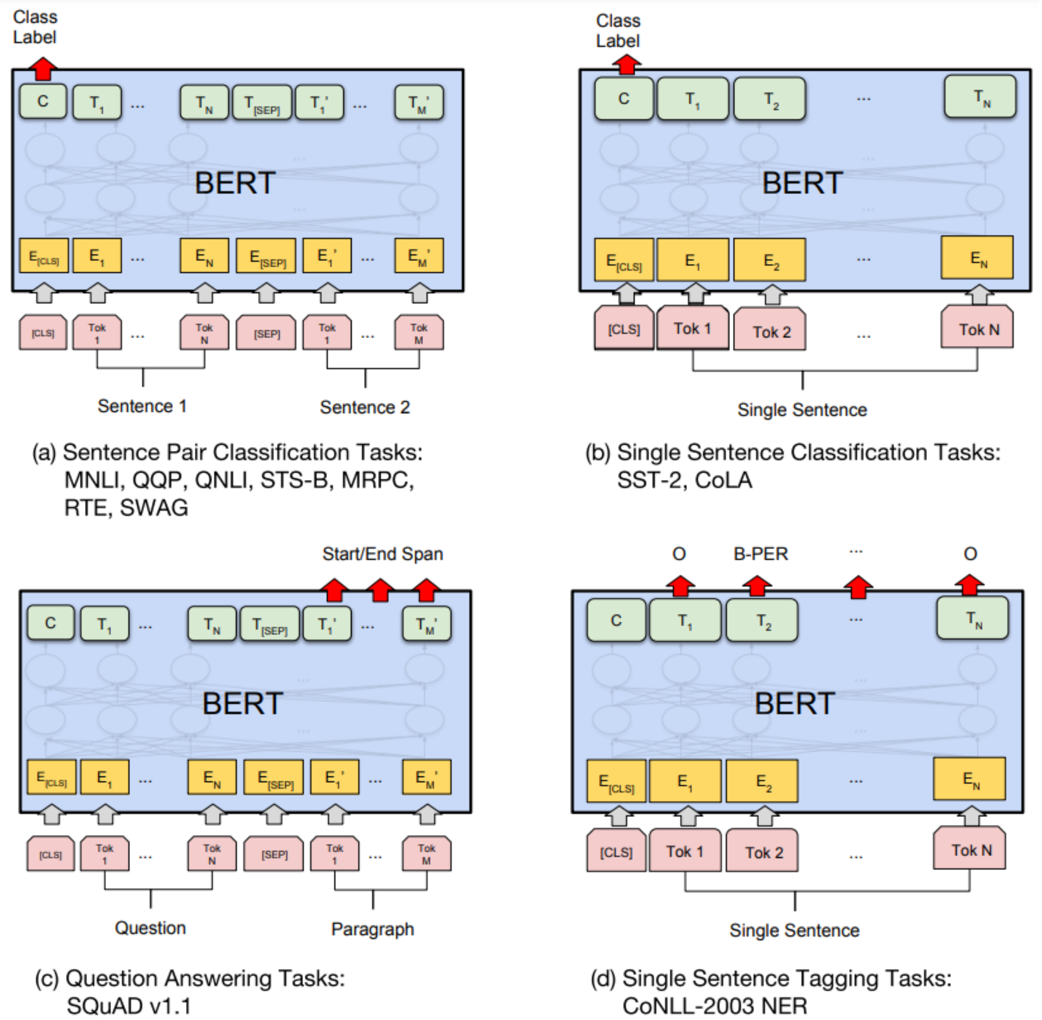


Fig. 2.4: Illustration of pre-training and fine-tuning of BERT on different tasks. Source: Devlin et al. [22]

Devlin et al. [22] used a tokenization algorithm called WordPiece [1] as BERT's vocabulary tokenizer. This algorithm is similar to the aforementioned BPE. WordPiece takes every character in the training dataset and merges these characters by creating symbol pairs



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Fig. 2.5: Detailed illustration of fine-tuning BERT on different tasks. Source: Devlin et al. [22]

based on the maximization of the likelihood of the training data, in case they will be added to the vocabulary.

BERT is based on the transformer architecture and its base version consists of 12 encoder blocks with 12 attention heads each and does not include decoder blocks. The large version of BERT consists of 24 encoder blocks with 16 attention heads each.

The procedure is similar to the transformer. The difference is that the input to the model is the result of adding the token embedding, the positional embedding, and an extra segment embedding, which denotes if the token belongs to the first or second sentence. BERT's pre-training is based on two objectives. The first one is the Next Sentence Prediction (NSP) and takes as input two sentences and tries to predict if the second sentence has a logical or sequential connection to the first one. The second task is called Masked Language Modeling (MLM) and masks a random token with a probability of 15%; the task is to predict which token was masked. It should be noted that each sequence starts with the token CLS, each sentence ends with the token SEP and the masked word is denoted with the token MASK.

RoBERTa

Liu et al. [46] from Facebook and Washington University, supported that Google's BERT was significantly under-trained and suggested RoBERTa, which stands for Robustly Optimized BERT pre-training approach. RoBERTa is based on BERT's architecture, but with modifications on key elements.

Firstly, the corpus that was used during the pre-training of RoBERTa included the one that BERT used (Book corpus and English Wikipedia). They also added CC-News articles, "Openwebtext" (web crawled content from Reddit) and "Stories" (data matched story-like style of Winograd NLP task). The whole corpus consisted of 161GB of text.

Regarding the tokenizer, Liu et al. [46] used a byte-level BPE instead of the WordPiece tokenizer following Radford et al. [54], who used bytes instead of unicode characters as the base subword units in BPE. This trick forces the base vocabulary to a size of 256 units, while every character is included in the vocabulary. Hence, this technique allows a sentence to be represented as a sequence of bytes, instead of characters. The advantage of this technique is that byte-level BPE can encode any text given, without the addition of the unknown token ('unk' symbol). Lastly, the vocabulary is consisted of 50K subword units, compared to BERT's 30K.

One of the main differences between RoBERTa and BERT was the removal of the Next Sentence Prediction objective (NSP). This means that the embeddings did not include the extra segment embeddings, as it was not needed to indicate if the token belonged in the first or the second sentence. Indeed, the removal of this objective in RoBERTa architecture lead to improvements in results compared to the ones in BERT.

Another improvement introduced by RoBERTa was the dynamic changing of the masking pattern. BERT used a static approach, that allowed the mask to be applied once in the data pre-processing. In RoBERTa, data were duplicated 10 times with 10 different masked patterns. Hence, the same mask was seen 4 times on 40 epochs of training.

Lastly, RoBERTa was pre-trained with bigger batch sizes and longer sequences. They trained the model for 500K steps with 8K batch size, compared to BERT's 1M steps with 256 batch size. These modifications improved optimization speed and increased performance.

Other Transformer-Based Models

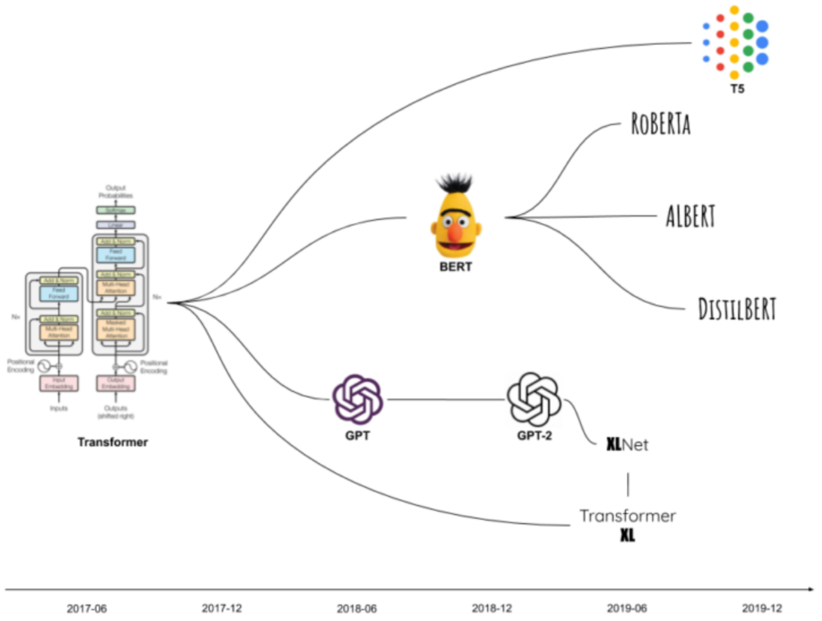


Fig. 2.6: Timeline of some Transformer-based models. There have been two main routes: masked-language models like BERT; and autoregressive models like GPT. Finally, the T5 deserves a special mention thanks to the text-to-text approach it proposes for multi-task learning in NLP. Source: `https://factored.ai/transformer-based-language-models/`

Figure 2.6 depicts a family tree timeline from transformer-based models. Specifically, it shows models across the years that were inspired from transformers and their architecture is based on them. Also, models that are based on BERT and are analyzed in this thesis can be seen.

There are many variations of the BERT. Lan et al. [42], introduced ALBERT (A Lite BERT), which is a more parameter efficient, lighter BERT. The three key techniques that were used consisted of splitting the embedding matrix into two smaller ones, sharing the parameters across layers ,and replacing of the Next Sentence Prediction (NSP) with a Sentence Order Prediction (SOP) task.

Sanh et al. [57] presented DistilBERT. They reduced the size and increased the speed of BERT, while keeping a comparable performance by the distillation of the pre-trained BERT. The NSP was not used during training. The model had 40% less parameters, it was 60% faster and it retained 95% of BERT's performance.

There are many other transformer-based models that tried to improve BERT's performance with different techniques (ConvBERT [36], ELECTRA [19], BART [44], Pegasus [68], tinyBERT [37], DeBERTa [31], BigBERT [67]) and even more are published until this day. Hence, this thesis will not dig deeper in the architecture of other models, as the most important ones are already analyzed. Based on the aforementioned models, we believe that RoBERTa is the most suitable for our research.

2.3 Domain-Specific Language Models

Recent research results showed that domain-specific models outperform generic models in domain-specific benchmarks. Lee et al. [43] created BioBERT by further pre-training BERT on biomedical corpora. In the same manner, Alsentzer et al. [3] further pre-trained BERT and BioBert on clinical notes, releasing Clinical BERT and Clinical BioBERT. Similarly, Beltagy et al. [7] pre-trained BERT on scientific publications and the resulting model was called SciBERT. SciBERT was either further pre-trained from BERT or pre-trained from scratch. Lastly, Loukas et al. [48] released SEC-BERT pre-trained on financial filings, performing better than FinBERT [66], which is a model pre-trained on financial domain. All these domain-specific models significantly outperformed the BERT-BASE model in most of their down-stream tasks respectively.

Chalkidis et al. [12, 11, 16] showed that BERT was the state-of-the-art in many legal NLP tasks. Chalkidis et al. [15] took their research one step further by testing strategies like using BERT out of the box, further pre-training BERT on legal corpora and pre-training BERT from scratch creating a legal oriented BERT called LEGAL-BERT. LEGAL-BERT achieved increased F1 scores in all down-stream tasks (binary classification on ECHR-Cases dataset, multi-label classification on ECHR-Cases and CONTRACTS-NER dataset). The research also included smaller models as the full capacity of larger and computationally more expensive models may be unnecessary in specialized domains, where syntax may be more standardized, the range of topics discussed may be narrower, terms may have fewer senses, etc. Lastly, Zheng et al. [69] suggest that domain over-specificity seems to be crucial on many occasions, creating CaseLawBERT (custom LEGAL-BERT). Domain pre-training improved scores in most legal NLU benchmark [17].

There are many other domain-specific pre-trained models. Some of them are presented in the following table 2.1:

2.4 Transformer-Based Multi-lingual Models

Most transformer-based models are focused on English. Later on, many mono-lingual models were pre-trained on different languages around the world. An alternative that many researchers tried to approach is to train multi-lingual models, which are capable of “learning” and adapting to several languages. These models also have the potential to adapt from one (source) language to others (targets) with limited or no training data

Model	Domain	Paper
FinBERT	Finance	Yang et al. [66]
SEC-BERT	Finance	Loukas et al. [48]
CaseLaw-BERT	Legal	Zheng et al. [69]
LEGAL-BERT	Legal	Chalkidis et al. [15]
RoBERTa-News	News	Gururangan et al. [29]
CodeBERT	Programming	Feng et al. [25]
Code-GPT (adapted)	Programming	Lu et al. [49]
PLBART	Programming	Ahmad et al. [2]
GraphCodeBERT	Programming	Guo et al. [28]
CoText	Programming	Phan et al. [53]
NetBERT	Networking	Louis [47]
OAG-BERT	Academic	Liu et al. [45]
SciBERT	Academic	Beltagy et al. [7]
MathBERT	Academic	Peng et al. [52]
TOD-BERT	Dialogue	Wu et al. [64]
BioBERT	Biomedical	Lee et al. [43]
ClinicalBERT	Biomedical	Alsentzer et al. [3]
PubMedBERT	Biomedical	Gu et al. [27]
HateBERT	Social Media	Caselli et al. [10]
RoBERTa-twitter	Social Media	Barbieri et al. [6]
RoBERTa-reviews	Social Media	Gururangan et al. [29]
BERT-SentiX	Social Media	Zhou et al. [70]
XLM-R-twitter	Social Media	Barbieri et al. [5]
BERTweet	Social Media	Nguyen et al. [51]
BERTweet-COVID19	Social Media	Nguyen et al. [51]
CT-BERT	Social Media	Müller et al. [50]

Tab. 2.1: A list of several domain-specific transformer-based language models with their respective papers. Many models in the list were gathered from Kalyan et al. [38]

in the target languages and without an absurd performance drop. These models have comparable results with mono-lingual models. This idea prevented the training of multiple mono-lingual models and the waste of resources.

Devlin et al. [22], released mBERT along with BERT. mBERT is a multi-lingual BERT that supports 104 languages. It was basically a model with the BERT architecture, but trained on a Wikipedia corpus that included 104 languages. To tackle the problem of more English articles than any other language, they under-sampled English and over-sampled the rest of the languages.

In 2019, Lample and Conneau [41], pre-trained a model similar to BERT, called XLM, with the Masked Language Modeling objective (MLM) and the casual language modeling objective (CLM) (next token prediction), but without the Next Sentence Prediction objective (NSP). Instead, they used a Translation Language Modeling objective (TLM), which takes as input the same sentence in two different languages concatenated and masks some tokens randomly, trying to predict the masked token by choosing tokens from the other language. The goal for the model is to learn similar representations of different languages. The model was pre-trained on a corpus of 15 different languages and outperformed mBERT in many tasks.

In November 2019, the Facebook AI team, inspired by RoBERTa, argued that other multi-lingual models like mBERT [22] and XLM [41] Masked Language Models are under-tuned. Hence, they released XLM-R, which is a multi-lingual version of the RoBERTa developed by Conneau et al. [20] to improve cross-lingual language understanding by studying the effects of training unsupervised cross-lingual representations at an enormous scale, resulting in training on one hundred languages. The XLM-R uses the Masked Language Modeling (MLM) objective, but unlike XLM it does not use the Translation Language Modeling (TLM) objective. The corpus used for training is consisted by text from CommonCrawl in 100 languages of a total 2.5TB. The vocabulary has a size of 250K compared to 50K of RoBERTa's vocabulary. The model obtained state-of-the-art performance on various cross-lingual NLP tasks and comparable results to RoBERTa in mono-lingual NLP tasks. (XLM-R-XL and XLM-R-XXL are XLM-R larger model variations [26]). It should be noted that XLM and XLM-R use larger architectures compared to mBERT, which may indicate that the difference in performance may not be completely accurate.

2.5 Evaluation Measures

The last subsection of this chapter presents the evaluation measures that will be used in the rest of this thesis for the evaluation of the models across several down-stream legal tasks.

Cross-entropy loss is a loss function that is used in machine learning and optimization. Cross-entropy measures the bits needed to transform the output probability distribution of a model for a specific input, to the actual distribution of that input. It measures the performance of a model. The loss score penalizes the probability of each predicted class based on the distance from the actual expected value. A higher cross-entropy loss means that the predictions diverge from the ground truth. Hence, loss should be as much as possible close to zero. It is also known as log loss or logarithmic loss. The cross-entropy loss can be defined as:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log_2 \hat{y}_i + (1 - y_i)\log_2 (1 - \hat{y}_i)\right] \tag{2.1}$$

where N is the total number of samples, y_i is the ground truth and $\hat{y}_i$ is the sample prediction.

MAE (Mean Absolute Error) is a loss function used in regression type problems. It is used when outliers should not be taken heavily into consideration and is defined as the absolute difference between the actual and the predicted values.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.2}$$

Accuracy is used during classification to measure the performance. It is defined as:

$$\text{Acc.} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{2.3}$$

Accuracy is not the ideal metric, when the classes are imbalanced, hence some other metrics need to be defined.

To evaluate the performance in multi-class classification, we use as metrics μ-F_1 (micro-F1) and m-F_1 (macro-F1). To define these metrics, we need first to define F1, Recall and Precision scores.

Recall is the number of correctly predicted positive samples, out of the number of all positive samples. It is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.4}$$

Precision is the number of correctly predicted positive samples, out of the total number of predicted positive samples. It is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.5}$$

where TP is the number of true positives, FP the number of false positives and FN the number of false negatives.

F1 score takes into consideration both Recall and Precision and provides useful insights about the performance of the model during classification. It is defined as:

$$\mathrm{F1} = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})} \tag{2.6}$$

F1 score provides information about the performance of one of the categories. During the multi-label classification, metrics that calculate the performance of all labels were used like:

m-F$_1$ (**macro-F1**) is the unweighted average of all F1 scores per label. It is defined as:

$$\mathrm{m{-}F_1} = \frac{\mathrm{F1_1} + \mathrm{F1_2} + ... + \mathrm{F1_C}}{C} \tag{2.7}$$

Where F1$_1$ is the F1 score of the first label and C is the number of labels.

μ-F$_1$ (micro-F1) is the global average F1 score, which is calculated by taking the sums of TP, FP, FN of each label. It is basically the proportion of correctly classified samples out of all the observation. Hence, it is equal to the accuracy score. It is defined as:

$$\mu{-}\mathrm{F_1} = \frac{\sum_{i=1}^{C} \mathrm{TP_i}}{\sum_{i=1}^{C} \mathrm{TP_i} + \frac{1}{2}\left(\sum_{i=1}^{C} \mathrm{FP_i} + \sum_{i=1}^{C} \mathrm{FN_i}\right)} \tag{2.8}$$

When the dataset is imbalanced, we need both of these metrics to understand how our models perform. μ-F$_1$ assigns weight to the labels with more samples, while m-F$_1$ treats all labels equally.

3 Model Specifications

Model Alias		#Langs	#Layers	#Units	#Heads	#Params	Vocab. Size	Train. Tokens
XLM-R	(base)	100	12	768	12	278M	250k	6.3T
XLM-R	(large)	100	24	1024	16	559M	250k	6.3T
C-XLM	(tiny)	10	4	128	4	9M	64k	92B
C-XLM	(small)	10	6	256	4	21M	64k	92B
C-XLM	(base)	10	12	512	8	71M	64k	92B
C-XLM	(large)	10	24	1024	16	368M	64k	92B

Tab. 3.1: Model Specifications, Training Tokens processed on pre-training for all variants of our XLM (C-XLM) models and the XLM-R models of Conneau et al. [20] considered as baselines.

This chapter presents all the specifications in the architecture of the models that were pre-trained. Following Chalkidis et al. [15], we pre-train from scratch legal domain-specific transformer-based language models. Our models are based on the RoBERTa architecture [46], i.e., trained with the Masked Language Modeling (MLM) objective, excluding the Next Sentence Prediction (NSP) one used by BERT [22]. Also, the dynamic changing of masking patterns and the better hyper-parameter tuning of RoBERTa produce better model performance.

In addition, based on the industry needs and driven by the work of Conneau et al. [20], our models are multi-lingual, which usually are referred as XLM in the literature. The models support ten languages in total (English, French, German, Greek, Spanish, Italian, Dutch, Polish, Portuguese, Russian).

Our hypothesis is that small, lightweight, and domain-specific models may perform well compared to the baselines.[1] We pre-train 4 variants of custom XLM models (C-XLM) starting from a large version with 24 Transformer blocks (layers), each consisting of 1024 hidden units and 16 attention heads and continue by decreasing each time by a factor of 2 across all dimensions, i.e., blocks/layers, hidden units, and attention heads. A minor exception in the tiny version, where we consider 4 attention heads of 32 hidden units per head instead of 2 attention heads with 64 units per head. All of these variations use GELU [33] as activation function, Adam optimizer, and a dropout rate of 0.1, similar to RoBERTa.

[1] XLM-R models are considered baselines. For more information about XLM-R see Sections 2.2 & 2.4

Lastly, it should be noted that C-XLM large has 191M parameters less than XLM-R large and C-XLM base has 207M parameters less than XLM-R base, while C-XLM small and C-XLM tiny have only 21M and 9M parameters respectively. One portion of the difference in size between our models and the baselines is due to the difference in number of languages (vocabulary size) each model uses, however our hypothesis suggests that even the smaller variants with way less parameters would be highly comparable in terms of performance with the baselines. Table 3.1 depicts the specifications of each model.

4 Pre-Training

This chapter presents the main components and the necessary steps of the pre-training, along with the results that the pre-trained models obtained. The first subsection describes the creation of the corpora that were used during training and the next one is referring to the tokenization and the creation of the custom vocabulary. Following, details about the pre-training and Masked Language Modelling (MLM) are discussed and finally the results of each variant of our model across languages are presented, along with a comparison to the baselines.

4.1 Training Corpora

We pre-trained our models using multi-lingual corpora that consist of regulations and contracts. For regulations, we used the MultiEURLEX dataset of Chalkidis et al. [14], published by the EU Publication Office,[1] that comprises 65k EU regulations officially translated in 24 languages. In our work, we consider 9 languages (English, French, German, Greek, Spanish, Italian, Dutch, Polish, Portuguese). We also considered additional publicly available English resources; specifically the 250 US code books, part of the "Pile of Law" corpus released by Henderson et al. [32], published by the U.S. Government Publishing Office,[2] alongside 36k UK laws (UK National Archives),[3] published by Chalkidis and Søgaard [18].

Regarding contracts, we considered the LEDGAR [62, 8] dataset comprising 900k sections from US contracts in English, published as exhibits in public filings at SEC-EDGAR[4] and 60k additional full contracts in English from a publicly available crawl from EDGAR.[5] As discussed in Henderson et al. [32], the content from these legal sources implicitly encodes privacy and toxicity rules since its content is handled by governments and courts, contrary to generic web material scraped from the web [23].

Since, there are no publicly available contracts in the rest of the languages, we translated these documents using state-of-the-art Neural Machine Translation (NMT) systems across

[1] https://eur-lex.europa.eu/
[2] http://www.gpo.gov/
[3] https://www.legislation.gov.uk/
[4] https://www.sec.gov/edgar/
[5] https://huggingface.co/datasets/albertvillanova/legal_contracts

all languages of interest with the addition of Russian. To achieve that, we used the OPUS-MT model from Helsinki-NLP [61] for the majority of the languages. Regarding languages that were not supported by OPUS-MT, we used an mBART model called mBART50_en2m from Facebook Research [60]. Both of these models were obtained from EasyNMT publicly available library.[6]

Table 4.1 displays the number of tokens that were included in contracts and regulations for each language. It should be noted that during the pre-training, 100% of regulations and 20% of translated contracts were used (100% of English contracts used). Hence, during the pre-training roughly 304M English tokens were used, while all the other languages consisted of approximately 90M-105M tokens (only 18M tokens were used from Russian due to the absence of Russian Regulations).

Corpus	**Tokens per Language**										
	EN	EL	DE	FR	ES	IT	NL	PL	PT	RU	All
Contracts	125.3M	111.1M	103.3M	121.7M	122.5M	113.4M	110.7M	89.8M	111.3M	89.6M	1.1B
Regulations	178.7M	71.4M	62.8M	74.0M	77.1M	70.1M	71.2M	31.9M	71.5M	-	708.7M
All	304M	182.5M	166.1M	195.7M	199.6M	183.5M	181.9M	121.7M	182.8M	89.6M	1.8B

Tab. 4.1: Total tokens used per language per pre-training corpus.

4.2 Custom Vocabulary

Relying on the above-mentioned resources, we built a custom vocabulary of 64k sub-word units that better fit the documents in the respective domains and languages of interest. We opted for Byte-Pair Encodings (BPEs) [58], similarly to most recent work on Transformer-based language models [55, 46, 20].

As it was previously mentioned in Section 2.2, the byte-level BPE uses bytes instead of Unicode characters as the base sub-word units in BPE. The BPE algorithm counts the frequency of the consecutive byte-pairs and merges the most frequent occurring byte-pairings for several iterations, until the vocabulary or the iterations reach the limit that was set. The advantage of this technique is that byte-level BPE can encode any text given, while the size of the vocabulary is 64K units. The max length of the tokenized text was set to 512 and the minimum frequency of a token in the corpus was 2. Lastly, the special tokens were <unk> for unknown words, <mask> for masked words in MLM, <s> denoting the beginning of a sentence, </s> denoting the end of a sentence, and <pad> a token that was padded in a sequence so that it reaches the designated maximum length.

[6] https://github.com/UKPLab/EasyNMT

4.3 Masked Language Modeling (MLM)

We pre-trained all variants of C-XLM (our domain-specific multi-lingual RoBERTa) for 1.1m steps (gradient updates) in total, based on a two-step approach, similarly to Devlin et al. [22], i.e., pre-train for 1m steps with sequences up to 128 sub-word units, followed by continued pre-training for 100k steps with sequences up to 512 sub-word units, always with a batch size of 512 sequences. This approach aims at a more efficient (compute-friendly) pre-training, since pre-training with shorter sequences severely decreases the needed compute and time.

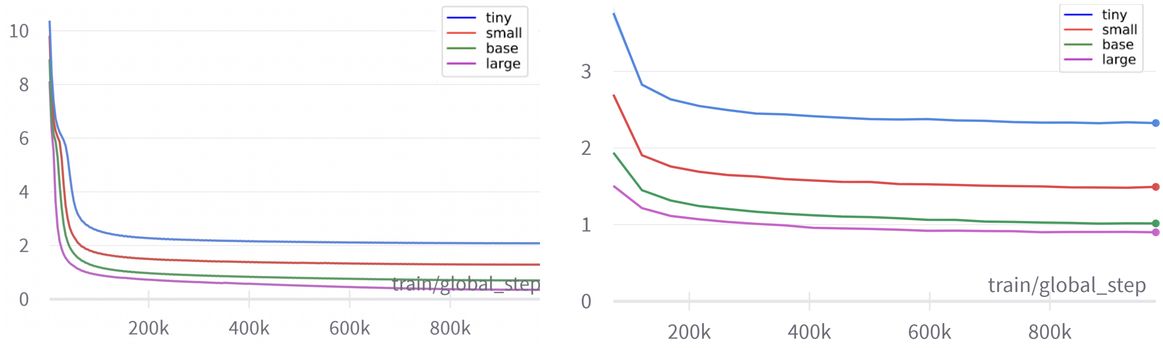
At each example during the Masked Language Modeling (MLM) objective, we mask out 15% of the tokens in total, trying to predict accurately the correct masked token. We pre-train all models with a warm-up in learning rate for the initial training steps (5% of 1M steps). Hence, the pre-training starts with a low, practically close to zero learning rate and then it increases over the first 50K steps, until it reaches the maximum learning rate of 1e-4, followed by a cosine decay. The exact same procedure is applied for the next 100K steps of pre-training, but the maximum learning rate is set to 1e-5.

In comparison, the XLM-R were pre-trained for 1.5m steps with batches of 8192 sequences, which accounts for approx. 63× more training tokens processed; the majority of those in high-resource languages like the ones we consider.

Lastly, it should be noted that for the pre-training of the models, four Google's Cloud TPUs v3-8 were used, each for a specific variant. The tiny version of C-XLM was pre-trained in total for approximately 4 days, the small version of C-XLM for 6 days, the C-XLM-base for 9 days and the larger version of C-XLM for 18 days.

4.4 MLM Results

In Figure 4.1, we observe the train and validation loss curves of differently sized models during pre-training. While models are equally poor performing in the very initial steps, larger models substantially outperform the smaller counterparts due to their increased capacity (number of parameters). All variants of our model seem not to be under-trained, as the train and validation curves flatten approaching the last training steps. The metrics seem to get better both in training and evaluation dataset almost until the final steps, but the rate of improvement seem to be exponential decreasing. Figure 4.1a depicts the training loss curves, while Figure 4.1b depicts the evaluation loss curves of the MLM. C-XLM large achieves an optimal training loss score of 0.35 with a validation loss score of 0.9. Similarly, C-XLM base achieves 0.69 and 1.03 in training and validation loss score respectively. Finally, C-XLM small and C-XLM tiny obtain higher loss scores as expected, with 1.28 and 2.08 in training and 1.53 and 2.37 in validation loss respectively.



(a) Train loss curves of C-XLMs.

(b) Evaluation loss curves of C-XLMs.

Fig. 4.1: Train and evaluation loss curves of C-XLMs during pre-training.

Table 4.2 presents the MLM accuracy our different models. More specifically, the MLM accuracy measures the correctly predicted masked tokens, relatively to the total number of masked tokens. As expected, the large version (81.5% overall accuracy) followed by the base version (77.9% overall accuracy) of C-XLM outperform their corresponding generic XLM-R models by 2.6% and 3.8% respectively. As expected, the small and tiny versions of C-XLM obtain worse results considering their limited capacity (68.9% and 54.9% respectively). However, it should be noted that a comparison between the XLM-R models of Conneau et al. [20] and our models (C-XLMs) is not ideal due to the different vocabulary used. Nevertheless, it provides a general idea on pre-training performance on legal specific corpora.

Figure 4.2 presents Masked Language Modeling performance in finer detail across languages per model, highlighting the predominance of our two largest models. The first and the second coloured webs depict the performance across languages in regulations and contracts, while the last one depicts the overall performance. A more fine-grained MLM evaluation (per language and per document type) can be found in Table 4.2. Regarding the performance across languages, our models (C-XLMs) consistently achieve the best results in French and Greek regulations, while French and English have the best results in contracts. Surprisingly, the baselines achieve their best results in Greek followed by French. Finally, according to Table 4.2, C-XLM base outperforms XLM-R large in many languages especially in contracts. Also, the small version of C-XLM achieves better results in English contracts (+0.75%) and is slightly outperformed by the base version of XLM-R, which is impressive given its much smaller size.

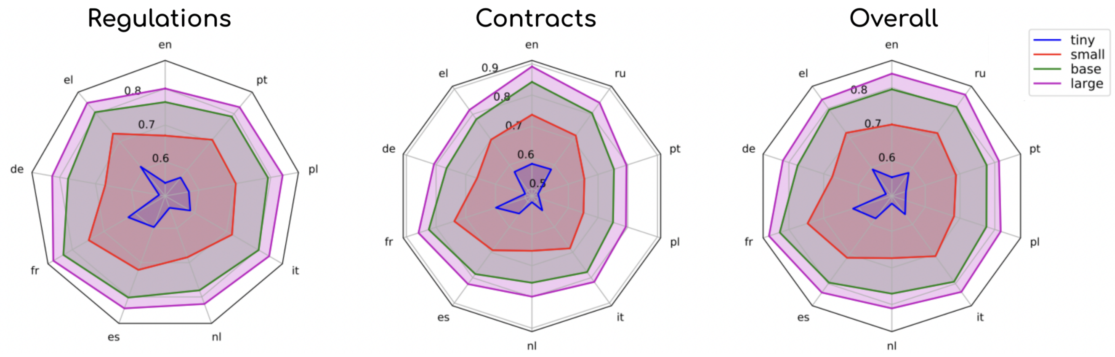


Fig. 4.2: MLM performance per language across C-XLM model variants depicted with different coloured webs.

Model	**C-XLM**								**XLM-R**			
Corpus Subset	**Tiny**		**Small**		**Base**		**Large**		**Base**		**Large**	
	Loss	**Acc.**	**Loss**	**Acc.**	**Loss**	**Acc.**	**Loss**	**Acc.**	**Loss**	**Acc.**	**Loss**	**Acc.**
Regulations (EN)	2.46	54.2%	1.62	67.1%	1.11	76.3%	0.97	80.0%	1.36	71.2%	1.07	76.3%
Regulations (EL)	1.89	61.0%	1.24	72.9%	0.85	80.5%	0.74	83.9%	0.93	79.2%	0.68	84.3%
Regulations (DE)	2.52	52.3%	1.60	67.3%	1.02	77.6%	0.84	82.1%	1.20	74.2%	0.89	80.0%
Regulations (FR)	1.84	62.3%	1.16	74.8%	0.75	82.8%	0.65	86.0%	1.01	77.8%	0.77	82.5%
Regulations (ES)	2.01	59.7%	1.31	72.1%	0.88	80.2%	0.78	83.3%	1.19	74.4%	0.93	79.0%
Regulations (NL)	2.43	54.0%	1.54	68.4%	1.00	78.1%	0.85	82.1%	1.25	73.8%	0.91	79.8%
Regulations (IT)	2.06	58.5%	1.32	71.8%	0.88	80.2%	0.77	83.5%	1.16	75.0%	0.88	80.2%
Regulations (PL)	2.23	57.2%	1.44	70.3%	0.94	79.2%	0.75	83.3%	1.08	76.8%	0.80	82.1%
Regulations (PT)	2.20	57.2%	1.43	70.7%	0.98	79.0%	0.87	82.3%	1.23	73.5%	0.94	78.8%
Regulations (All)	2.36	55.1%	1.53	68.9%	1.03	77.7%	0.90	81.4%	1.18	74.6%	0.90	79.8%
Contracts (EN)	1.96	58.0%	1.15	73.7%	0.66	84.2%	0.45	89.1%	1.24	73.0%	0.93	78.8%
Contracts (EL)	2.11	57.2%	1.41	70.0%	0.99	78.0%	0.84	81.8%	1.07	76.8%	0.87	80.9%
Contracts (DE)	2.62	50.0%	1.65	66.2%	1.07	76.5%	0.89	80.7%	1.27	73.0%	1.04	77.4%
Contracts (FR)	1.99	59.9%	1.23	73.9%	0.78	82.5%	0.65	85.9%	1.09	76.8%	0.88	80.7%
Contracts (ES)	2.28	54.7%	1.45	69.2%	0.95	78.6%	0.80	82.5%	1.32	72.4%	1.08	76.4%
Contracts (NL)	2.64	49.7%	1.69	65.2%	1.12	75.5%	0.93	79.9%	1.43	70.6%	1.15	75.5%
Contracts (IT)	2.37	53.4%	1.52	68.4%	1.01	77.8%	0.86	81.7%	1.38	71.5%	1.14	75.9%
Contracts (PL)	2.66	50.2%	1.74	65.1%	1.17	75.0%	0.98	79.4%	1.14	75.3%	0.94	79.5%
Contracts (PT)	2.62	49.8%	1.69	65.4%	1.14	75.4%	0.96	79.6%	1.67	66.8%	1.46	70.2%
Contracts (RU)	1.98	58.2%	1.29	71.6%	0.85	80.5%	0.70	84.6%	1.18	75.4%	0.99	78.8%
Contracts (All)	2.26	56.6%	1.43	71.0%	0.97	79.5%	0.87	82.9%	1.41	71.7%	1.16	76.1%
Overall (EN)	2.28	55.5%	1.44	69.9%	0.93	79.5%	0.76	83.8%	1.32	71.9%	1.02	77.3%
Overall (EL)	2.02	59.4%	1.33	71.7%	0.92	79.5%	0.81	82.9%	1.04	77.5%	0.81	82.1%
Overall (DE)	2.53	52.1%	1.60	67.5%	1.04	77.5%	0.87	81.7%	1.27	73.4%	0.99	78.6%
Overall (FR)	1.92	61.6%	1.20	74.6%	0.78	82.6%	0.69	85.7%	1.09	76.8%	0.86	81.2%
Overall (ES)	2.12	57.9%	1.36	71.2%	0.92	79.6%	0.81	83.0%	1.28	73.1%	1.03	77.6%
Overall (NL)	2.52	52.4%	1.60	67.4%	1.06	77.1%	0.91	81.1%	1.37	72.0%	1.06	77.4%
Overall (IT)	2.20	56.6%	1.40	70.7%	0.94	79.3%	0.83	82.8%	1.30	72.8%	1.03	77.8%
Overall (PL)	2.41	54.3%	1.56	68.3%	1.04	77.6%	0.86	81.7%	1.15	75.9%	0.90	80.5%
Overall (PT)	2.37	54.4%	1.53	68.8%	1.05	77.7%	0.93	81.2%	1.47	69.9%	1.22	74.4%
Overall (RU)	1.98	58.2%	1.29	71.6%	0.85	80.5%	0.70	84.6%	1.18	75.4%	0.99	78.8%
Overall	2.37	54.9%	1.53	69.0%	1.03	77.9%	0.90	81.5%	1.23	74.0%	0.96	79.0%

Tab. 4.2: MLM Validation Performance Scores (Cross-Entropy Loss, Accuracy) for our C-XLM and XLM-R models.

5 Fine-Tuning

This section presents the fine-tuning process of the different variants of our C-XLM models. We use 5 different down-stream legal NLP tasks, comprising both publicly available and private English and multi-lingual datasets. The task types are document and sentence classification, natural language inference and entity extraction. Following, the experimental setup of each down-stream task will be analyzed. Finally, the results will be presented for each task in detail, in the form of comparisons in performance between languages and between C-XLM variants and the XLM-R baselines.

5.1 Benchmark - Tasks and Datasets

In this sub-section, we briefly present the evaluation benchmark that we use, which consist of both publicly available and private datasets. The benchmark is diverse covering three task types (document, sentence, and token classification) and two multi-lingual datasets.[1] The datasets in detail are:

MultiEURLEX [13], a multi-lingual dataset for legal topic classification comprising 65k EU laws officially translated in 23 EU languages.[2] Each document (EU law) was originally annotated with relevant EUROVOC[3] concepts by the Publications Office of EU. We use the 21 'Level 1' labels, obtained by Chalkidis et al. [13] from the original EUROVOC annotations of the documents. We use a derivative of the original dataset considering only 1k non-parallel documents per supported language (9k in total for 9 languages). This is inline with the work of Xenouleas et al. [65], where the authors consider a more "realistic" harder version of MultiEURLEX with less and non-parallel documents. This is a multi-label document classification task; thus we evaluate performance using macro- (m-F_1) and micro- (μ-F_1) F1 scores.

[1] We do not use the LexGLUE benchmark of Chalkidis et al. [17], since it is monolingual (English only) and also covers tasks that involve litigation, which are out of scope.

[2] MultiEURLEX is available at `https://huggingface.co/datasets/multi_eurlex`.

[3] EUROVOC is a hierarchically organized taxonomy of concepts (a hierarchy of labels) available at `http://eurovoc.europa.eu/`.

UNFAIR-ToS [24] is a dataset for detecting unfair clauses in Terms of Service (ToS) agreements from online platforms (e.g., YouTube, Facebook, etc.) in 4 languages (English, German, Italian, and Polish). The dataset consists of 2074 training samples and has been annotated on the sentence-level with 8 types of *unfair contractual terms*, meaning terms (sentences) that potentially violate user rights according to EU consumer law. Sentences have been also annotated according to a 3-level fairness score (*fair*, *partially unfair*, *clearly unfair*). In our case, we examine the latter task as sentence regression and evaluate performance using Mean Absolute Error (MAE), and Accuracy (Acc.) on rounded (discrete) scores.

ContractNLI [39] is a dataset for contract-based Natural Language Inference (NLI). The dataset consists of 607 contracts, specifically Non-Disclosure Agreements (NDAs). Each document has been paired with 17 templated *hypotheses* and labeled with one out of three classes (*entailment*, *contradiction*, or *neutral*). We examine a lenient version of this task, where instead of the full document (NDA), we represent the document with a short number of sentences which have been annotated as rationales for the specific task. The train sample consists of 6819 train samples. This is a single-label multi-class document classification task and we evaluate performance using macro- (m-F_1) and micro- (μ-F_1) F1 scores.

Contract-Obligations is a proprietary (privately developed) dataset for obligation extraction from contracts (legal agreements). The dataset consists of N contracts of several types (service, employment, purchase, etc.). Each contract has been split into paragraphs, and has been labeled with 4 obligation sub-types, i.e., *Obligation*, *Deliverable*, *Discretion*, and *Prohibition*, while some paragraphs are not relevant, resulting in a total of 5 potential classes. This is a single-label multi-class document classification task. We evaluate performance using macro- (m-F_1) and micro- (μ-F_1) F1 scores.

ContractNER is a proprietary (privately developed) dataset for contract element extraction. The dataset consists of N contract introductions from several types (service, employment, purchase, etc.) of contracts. Each introduction (paragraph) has been labeled with 4 entity types (*Title*, *Contracting Party*, *Start Date*, *Effective Date*). This is a single-label multi-class token classification task. Thus, we evaluate performance using macro- (m-F_1) and micro- (μ-F_1) F1 scores on entity level.

In another note, many of these sources that we used to pre-train our C-XLM models, overlap with the benchmark datasets we used to evaluate the very same models, e.g., the MultiEURLEX dataset used both for pre-training and evaluation (Sections 4.1 and 5.1). This is inline with Krishna et al. [40], who recently showed that down-stream datasets should be used as up-stream (pre-training) corpora, if domain specificity and such applications is the goal, in contrast to heavy generalization across domains and acquirement of common knowledge. Of course, considering fair evaluation practices, we do not use the test subsets of the down-stream tasks during pre-training.

Table 5.1 contains all the necessary information of the five datasets, including the source, the sub-domain, the number of languages, the type of task, the number of samples and the number of classes of each dataset. In addition, Table 5.2 depicts examples of some samples for publicly available datasets along with their designated labels.

Dataset	Source	Sub-domain	#Languages	Task Type	Train/Dev/Test Samples	#Classes
MultiEURLEX	[13]	EU Laws	9	Multi-label Classification	9000/900/900	21 'Level 1'
UNFAIR-ToS	[24]	ToS agreements	4	Regression	2074/191/417	3
ContractNLI	[39]	Contracts (NDAs)	1	Multi-class Classification	6819/978/1991	3
Contract-Obligations	Private	Contracts	1	Multi-class Classification	Private	5
ContractNER	Private	Contracts	1	Multi-class Classification	Private	9

Tab. 5.1: The table provides information about the 3 publicly available and 2 private datasets. Details like the source, the sub-domain, the number of languages, samples and classes are provided. In addition, the task type is depicted. It should be noted that the task type of UNFAIR-ToS is regression, but the accuracy score is provided through rounded (discrete) scores.

Dataset	Sample	Label
MultiEURLEX	'Commission Regulation (EC) No 54/2003 of 13 January 2003 establishing the standard import values for determining the entry price of certain fruit and vegetables THE COMMISSION OF THE EUROPEAN COMMUNITIES, Having regard to the Treaty establishing the European Community ,regard to Commission Regulation (EC) No 3223/94 of 21 December 1994 on detailed rules for the application of the import arrangements for fruit and vegetables(1) , as last amended by Regulation (EC) No 1947/2002(2),...'	'2', '17', '6'
UNFAIR-ToS	'Diese allgemeinen Geschäftsbedingungen, die gelegentlichen Veränderungen unterliegen, gelten für alle unsere Dienstleistungen, die unmittelbar oder mittelbar (d.h. über Dritte) über das Internet, jegliche Art von mobilen Endgeräten, per E-Mail oder per Telefon zur Verfügung gestellt werden.'	'potentially_unfair'
ContractNLI	'premise': '5. All Confidential Information in any form and any medium, including all copies thereof, disclosed to the Recipient shall be returned to UNHCR or destroyed: (a) if a business relationship is not entered into with UNHCR on or before the date which is three (3) months after the date both Parties have signed the Agreement; or ' ,'hypothesis': 'Receiving Party shall destroy or return some Confidential Information upon the termination of Agreement.'	'Entailment'
Contract-Obligations	Private	'Deliverable'
ContractNER	Private	Sequence of ('B-TITLE',...,'B-PARTY',...,'B-EFDATE',...)

Tab. 5.2: Examples from random samples along with their label for each one of the public datasets.

5.2 Experimental Set Up

Devlin et al. [22] suggested a hyper-parameter tuning, that was adopted in many papers [43, 7, 3, 59]. This hyper-parameter tuning included a light grid-search in learning rate ∈ {2e-5, 3e-5, 4e-5, 5e-5}, the number of training epochs ∈ {3, 4}, and the batch size ∈ {16, 32} with a fixed dropout rate of 0.1. In our research, we tune each variation of our model based on a grid-search of learning rate on the following range ∈ {1e-4, 3e-4, 1e-5, 3e-5, 5e-5, 1e-6}. The batch size is fixed to 16, and the dropout rate at 0.1. The max sequence length is fixed to 512 for MultiEURLEX and ContractNLI, 256 for ContractNER and 128 for Contract-Obligations and UNFAIR-ToS based on the training subset statistics. Lastly, Chalkidis et al. [15] found that some models may under-fit for 4 epochs. Hence, following their work, we use early stopping based on validation loss up to 20 maximum train epochs with a patience of 3 epochs. We select and report test scores based on the model with the best validation performance.

Table 5.3 displays the optimal learning rates for each model variation and baseline that were used during the fine-tuning process for every different task. Each one was selected through grid-search, between the learning rates that were previously mentioned. We observe that smaller models favor larger learning rates, i.e., 1e-4 and 5e-5 in most cases, while larger models favor smaller learning rates, i.e., 1e-5 and 3e-5. It should be noted that different learning rates in each task and variant of C-XLM, usually resulted in a big difference in performance. Thus, we had to be careful in the choice of learning rate and thoroughly examined a wide range of learning rates through grid-search.

Model	Alias	MultiEURLEX	UNFAIR-ToS	CNLI	Obligations	ContractNER
XLM-R	(Base)	1e-5	3e-5	1e-5	1e-5	5e-5
XLM-R	(Large)	3e-5	1e-5	1e-5	1e-6	5e-5
C-XLM	(Tiny)	1e-4	3e-4	1e-4	1e-4	1e-4
C-XLM	(Small)	1e-4	5e-5	1e-4	5e-5	5e-5
C-XLM	(Base)	5e-5	3e-5	5e-5	3e-5	3e-5
C-XLM	(Large)	5e-5	5e-5	3e-5	1e-5	5e-5

Tab. 5.3: Optimal Learning Rates per down-stream task across all variants of C-XLM and XLM-R models. Note: CNLI refers to ContractNLI and Obligations to Contract-Obligations tasks.

5.3 Fine-tuning Results

Table 5.4 presents the results of the fine-tuned baselines, XLM-R models, (upper zone) and of all the variants of our C-XLM models (lower zone) for each down-stream task. We hypothesize that the base and large versions of C-XLM will perform better compared to their counterpart XLM-R models and that smaller version of C-XLM models will be

Model	Alias	MultiEURLEX		UNFAIR-ToS		CNLI		Obligations		ContractNER	
		μ-F_1	m-F_1	Acc.	MAE	μ-F_1	m-F_1	μ-F_1	m-F_1	μ-F_1	m-F_1
XLM-R	(base)	75.3	53.2	86.6	0.17	84.0	81.9	89.7	88.2	92.4	93.9
XLM-R	(large)	77.8	63.8	89.0	0.16	**86.3**	**84.7**	88.9	87.4	92.8	93.7
C-XLM	(tiny)	66.5	46.1	78.2	0.27	70.2	69.2	88.7	87.4	87.2	89.3
C-XLM	(small)	72.3	54.7	85.4	0.20	79.7	77.0	90.4	89.0	90.1	92.4
C-XLM	(base)	75.3	59.4	87.3	0.18	84.0	82.1	91.2	90.4	92.9	93.9
C-XLM	(large)	**78.4**	**65.4**	**89.7**	**0.14**	85.3	83.0	**91.8**	**90.6**	**93.2**	**94.6**

Tab. 5.4: Overall results our fine-tuned C-XLM models and the baseline XLM-R moodels across all down-stream tasks. Note: CNLI refers to ContractNLI and Obligations to Contract-Obligations tasks.

highly comparable to the XLM-R base version. Indeed, the base version of C-XLM always outperforms XLM-R across all 5 datasets, while the large version of C-XLM outperforms XLM-R in all but one (4 out of 5) datasets (ContractNLI). Also, the small version of C-XLM model obtains comparable results to the base version of XLM-R, which is impressive considering its 13× smaller size. Surprisingly, it even outperforms the baseline model in Contract-Obligation dataset.

The rest of this chapter provides information with greater detail considering the obtained results. Results for each one of the datasets are presented separately and finally a deeper dive into the multi-lingual datasets MultiEURLEX and UNFAIR-ToS will be presented, regarding the performance per language.

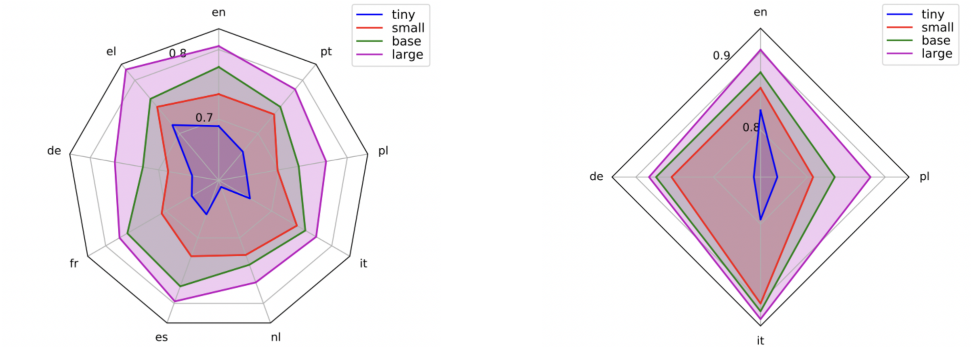
MultiEURLEX: Both large versions of C-XLM and XLM-R clearly outperform the rest of the models with the C-XLM outperforming XLM-R by 0.6 p.p. in μ-F_1 and 1.6 p.p. in m-F_1. Similarly, the base version of C-XLM outperforms the equivalent version of XLM-R in terms of m-F_1 impressively by 6.2 p.p. However, they obtain similar scores in μ-F_1. Interestingly, the small version of C-XLM has comparable performance with the base version of XLM-R while being approx. 13× smaller, surprisingly outperforming it by 1.5 p.p. in terms of m-F_1.

UNFAIR-ToS: Both large and base versions of C-XLM outperform their counterpart XLM-R models by 0.7 p.p. in accuracy, although the mean absolute error in the base version of C-XLM is surprisingly slightly higher than the corresponding XLM-R by 0.01. Again, the small version of C-XLM achieves competitive performance to base-sized models. However, the tiny version of our C-XLM seems to slightly underperform relatively to its counterparts and relatively to other tasks.

ContractNLI: In this task, we find that the large version of XLM-R outperforms the one of C-XLM (+1 p.p. in μ-F_1 and +1.7 p.p. in m-F_1), while both base models perform comparably, having the same μ-F_1 and 0.2 p.p. difference in m-F_1 in favor of C-XLM base model. We also note that the relative differences between differently sized models are the more intense across all tasks. The small version of C-XLM has more than 8 p.p. higher μ-F_1 and m-F_1 compared to tiny version, while the base version outperforms the small one in both metrics by 4.3 p.p. and 5.1 p.p. respectively.

Contract-Obligations: On this task, all variants of C-XLM model, except the tiny version, outperform the baselines (XLM-R). Specifically, the large version of C-XLM achieves +2.9 p.p. in μ-F_1 and +3.2 p.p. in m-F_1 compared to the large version of XLM-R. Similarly, the base version of C-XLM achieves better results than the baseline XLM-R base by 1.5 p.p in μ-F_1 and 2.2 p.p. in m-F_1. Oddly enough, the base version of the baseline XLM-R slightly outperforms the large version. However, comparing the small version of C-XLM with the highest performing baseline, we obtain results that favor our small model by 0.7 p.p in μ-F_1 and by 0.8 p.p in m-F_1, while being approx. 13× smaller. Lastly, the tiny version of C-XLM achieves impressive results and is slightly outperformed by the baseline by 1 p.p in μ-F_1 and 0.8 p.p in m-F_1, while being approx. 31× smaller.

ContractNER: Similarly, our C-XLM models outperform the corresponding large and base baselines by approx. 0.5 p.p. in μ-F_1. In addition, m-F_1 is higher in our large model by 0.9 p.p., while base models have identical results. Again, the small version of C-XLM is competitive to the baseline.



(a) MultiEURLEX

(b) UNFAIR-ToS

Fig. 5.1: Radar plots with per language performance for the multi-lingual MultiEURLEX and Unfair-ToS datasets for all the versions of C-XLM.

Language Parity: Figure 5.1 provides information through radar plots, about scores on MultiEURLEX and UNFAIR-ToS datasets per language for each variant of C-XLM. We generally observe that performance varies across languages.

In the MultiEURLEX dataset, models perform better in Greek, followed by Spanish and English, while German obtain the worst overall results for the large version of C-XLM.

This does not apply for the other versions as language performance disparity varies across different size models, depicted as differently shaped webs.

On the other hand, in UNFAIR-ToS dataset, the models perform better in Italian followed by English, which for example is the opposite relatively to MultiEURLEX dataset. In this particular task, there is not so much variety in language performance disparity, as the webs have similar shape, apart from the tiny version of C-XLM.

MultiEURLEX per Language μ-F$_1$

Model	**Alias**	EN	FR	DE	NL	IT	ES	PT	PL	EL	μ-F$_1$
XLM-R	(large)	80.6	78.8	76.1	76.1	78.5	80.1	75.7	75.8	78.7	77.8
XLM-R	(base)	77.6	76.9	73.7	74.3	76.3	75.0	75.7	73.2	75.6	75.3
C-XLM	(large)	80.5	77.7	76.4	76.8	77.4	79.7	78.3	76.9	82.0	78.4
C-XLM	(base)	77.5	76.4	72.3	74.1	75.6	77.4	75.0	72.9	76.5	75.3
C-XLM	(small)	73.6	70.7	68.6	72.6	74.2	72.8	73.6	69.8	75.0	72.3
C-XLM	(tiny)	69.0	65.7	65.1	62.2	66.4	66.4	66.6	65.3	71.6	66.5

Tab. 5.5: Micro-F1 (μ-F$_1$) scores for the MultiEURLEX task per language. Results across all variants of our model C-XLM and the baseline XLM-R are reported.

MultiEURLEX per Language m-F$_1$

Model	**Alias**	EN	FR	DE	NL	IT	ES	PT	PL	EL	m-F$_1$
XLM-R	(large)	64.3	67.3	60.3	61.9	58.1	63.6	57.1	60.0	63.5	63.8
XLM-R	(base)	56.2	54.7	50.9	53.6	49.5	52.4	52.2	49.8	52.0	53.2
C-XLM	(large)	66.8	63.2	63.0	63.9	55.0	67.1	60.6	61.6	70.7	65.4
C-XLM	(base)	59.6	58.8	55.1	59.7	54.0	59.1	59.8	57.1	60.3	59.4
C-XLM	(small)	55.9	52.7	50.8	55.8	52.6	57.0	55.6	50.8	56.2	54.7
C-XLM	(tiny)	50.1	43.7	47.7	42.6	38.3	46.8	47.6	41.9	44.1	46.1

Tab. 5.6: Macro-F1 (m-F$_1$) scores for the MultiEURLEX task per language. Results across all variants of our model C-XLM and the baseline XLM-R are reported.

Diving into greater detail, Table 5.5 and Table 5.6 present the μ-F$_1$ and m-F$_1$ scores per language for MultiEURLEX dataset respectively. We observe that the baselines give an edge in performance for the English language, compared to C-XLM models that perform better in Greek. In terms of μ-F$_1$, even though our large version of C-XLM outperforms the large version of XLM-R overall, our model outperforms the baseline in German, Dutch, Portuguese, Polish and Greek (English, French, Italian, Spanish have slightly better performance in XLM-R). However, in terms of m-F$_1$, our model performs better in most languages (underperforms only in French by 4.1 p.p. and Italian by 3.1 p.p.), compared to the baseline. Regarding the base version of C-XLM, our model is slightly outperformed by the base version of XLM-R in most languages, except for Greek and Spanish (which is surprising as the results contradict in the large version) in terms of μ-F_1. On the other hand, we observe a relatively big gap in performance in favor of our model in all languages in terms of m-F_1. Finally, it is impressive that the small variant of C-XLM outperforms the base version of XLM-R in terms of m-F_1 in all languages except for English and French. However, this is not the case in terms of μ-F_1, where C-XLM small is clearly outperformed.

UNFAIR-ToS per Language Scores

Model	Alias	EN		PL		IT		DE		Total	
		MAE	Acc.	MAE	Acc.	MAE	Acc.	MAE	Acc.	MAE	Acc.
XLM-R	(large)	0.16	89.3	0.19	87.2	0.14	91.2	0.15	88.3	0.16	89.0
XLM-R	(base)	0.14	90.3	0.22	81.7	0.15	88.2	0.17	86.4	0.17	86.6
C-XLM	(large)	0.13	90.3	0.17	88.1	0.11	92.2	0.15	88.3	0.14	89.7
C-XLM	(base)	0.17	87.4	0.22	83.5	0.13	91.2	0.19	87.4	0.18	87.3
C-XLM	(small)	0.17	85.4	0.24	80.7	0.16	90.2	0.21	85.4	0.20	85.4
C-XLM	(tiny)	0.27	82.5	0.28	76.1	0.24	79.4	0.30	74.8	0.27	78.2

Tab. 5.7: Mean Absolute Error (MAE) and Accuracy (Acc.) scores for the UNFAIR-ToS task per language. Results across all variants of our model C-XLM and the baseline XLM-R are reported.

Lastly, Table 5.7 includes the Mean Absolute Error and Accuracy scores per language for the UNFAIR-ToS dataset. In this task, the performance for the large variant of C-XLM is as expected greater in every language (German obtains the same exact results) in both evaluation measures compared to the XLM-R large. Regarding the base version of C-XLM, it achieves better accuracy in Polish, Italian and German, although the Mean Absolute Error is smaller in the latter language compared to the baseline's base version. The baseline outperforms our model only in English. Finally, our small variant of C-XLM obtains impressive results for its size and also outperforms the base version of XLM-R in the Italian language.

We cross out representation disparity as a possible explanation, since training data equally represent all languages (equal number of training examples). Interestingly, pre-training (MLM) accuracy also does not correlate with the down-stream performance. We expected that the best performance would be achieved in French, followed by English, Spanish, and English in that order for MultiEURLEX and English, followed by Italian for UNFAIR-ToS, which is not the case. Based on the aforementioned points, we can only hypothesize that other qualitative characteristics, such as idiosyncrasies of a language in a specific context/domain, are responsible for performance disparities in-between languages.

In general trends, we observe that larger models outperform smaller ones in most cases, and domain-specific models outperform generic ones, while using a substantially smaller (4×) vocabulary and being significantly less (63×) pre-trained. The largest relative differences occur in MultiEURLEX, a 20-class multi-label classification task, and ContractNLI, a sentence pair classification task. Also, the best results are achieved in Contract-Obligations task.

6 Conclusions

Summarizing, based on Chalkidis et al. [15], we wanted to pre-train a multi-lingual RoBERTa-based model on domain-specific legal corpora. The initial assumption was that a huge legal domain-specific corpus, consisting of 10 languages, along with a model that is based on RoBERTa architecture [43, 20] would achieve state-of-the-art results in several down-stream tasks.

The goal was to provide useful insights on model development, so that a tech company/start-up could adapt to new technological trends, achieve state-of-the-art results and probably adjust their strategy in a way that they would save large compute resources that have economic consequences.

We started from ground zero and pre-trained a RoBERTa model from scratch in a corpus comprising legal regulations and contracts. We created four variants of this model, each one with different size. The results suggested that the larger models outperformed the smaller ones in the Masked Language Modelling (MLM) objective. It should be noted that different sized C-XLMs achieved different top scores in different languages and document types, which can be confirmed from the different shape of coloured webs in Figure 4.2. In addition, our two biggest models outperformed the corresponding baselines (although a comparison may be unfair due to different vocabulary used for pre-training). Lastly, the small C-XLM had competitive performance to the base variant of XLM-R.

The following steps were to fine-tune our pre-trained C-XLM models in five different down-stream tasks. We benchmarked their performance in three public (two of them are multi-lingual) and two private datasets, consisted of tasks like legal document classification (MultiEURLEX), sentence classification/regression (Contracts-Obligations, UNFAIR-ToS), natural language inference (ContractNLI) and entity extraction (ContractNER). The results suggested that larger domain-specifc models outperform smaller ones and that our domain-specific C-XLM models outperform their corresponding generic XLM-R models, even though they use smaller vocabulary and are less pre-trained. Also, the base version of C-XLM outperforms the baseline in all 5 tasks, while the large one does in four out of five tasks. Lastly, the small version of C-XLM achieves results close or even better to the base version of XLM-R, although it is 13 times smaller.

Concluding, based on the above findings, our research questions are validated. Indeed, the results can be summarized in three general points:

1. Multi-lingual, legal domain-specific, RoBERTa-based models [43] outperform the multi-lingual XLM-RoBERTa models [20] in legal NLP tasks, although gains are decreased considering much large models.

2. The larger language models significantly outperform smaller ones; the performance increase varies across tasks.

3. The small versions of our language models perform better/similar to the base version of XLM-R.

Regarding future work ideas, further domain-specific pre-training strategies should be explored. Also, the performance can be further explored by incorporating more tasks in legal NLP. In addition, it would be interesting to approach concepts like pruning/ distillation/ quantization that are used to compress a model and rate the trade-off between compression and performance/ efficiency. This way a guideline for tech companies and Legal-Tech practitioners could be created that provides input with regards to whether a specific-sized model should be pre-trained from scratch or if the largest model possible should be pre-trained, followed by compression techniques to achieve the optimal results, saving both time and resources that have huge economic consequences.

Bibliography

[1] *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012.* IEEE, 2012.

[2] Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. *Unified Pre-training for Program Understanding and Generation.* 2021.

[3] Emily Alsentzer, John Murphy, William Boag, et al. “Publicly Available Clinical BERT Embeddings”. In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop.* Minneapolis, Minnesota, USA: Association for Computational Linguistics, June 2019, pp. 72–78.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization.* 2016.

[5] Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. *XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond.* 2021.

[6] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. “TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020.* Online: Association for Computational Linguistics, Nov. 2020, pp. 1644–1650.

[7] Iz Beltagy, Kyle Lo, and Arman Cohan. “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).* Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3615–3620.

[8] Łukasz Borchmann, Dawid Wisniewski, Andrzej Gretkowski, et al. “Contract Discovery: Dataset and a Few-Shot Semantic Retrieval Challenge with Competitive Baselines”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020.* Online: Association for Computational Linguistics, Nov. 2020, pp. 4254–4268.

[9] Tom Brown, Benjamin Mann, Nick Ryder, et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems.* Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[10] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. *HateBERT: Retraining BERT for Abusive Language Detection in English.* 2020.

[11]Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. "Neural Legal Judgment Prediction in English". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4317–4323.

[12]Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. "Large-Scale Multi-Label Text Classification on EU Legislation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 6314–6322.

[13]Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. "MultiEURLEX - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online, Nov. 2021.

[14]Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. "MultiEURLEX - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6974–6996.

[15]Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. "LEGAL-BERT: The Muppets straight out of Law School". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online, 2020, pp. 2898–2904.

[16]Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. "Neural Contract Element Extraction Revisited: Letters from Sesame Street". In: (2021).

[17]Ilias Chalkidis, Abhik Jana, Dirk Hartung, et al. "LexGLUE: A Benchmark Dataset for Legal Language Understanding in English". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4310–4330.

[18]Ilias Chalkidis and Anders Søgaard. "Improved Multi-label Classification under Temporal Concept Drift: Rethinking Group-Robust Algorithms in a Label-Wise Setting". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2441–2454.

[19]Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. *ELECTRA: Pretraining Text Encoders as Discriminators Rather Than Generators*. 2020.

[20]Alexis Conneau, Kartikay Khandelwal, Naman Goyal, et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451.

[21]Alexis Conneau and Guillaume Lample. "Cross-lingual Language Model Pretraining". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, et al. Vol. 32. Curran Associates, Inc., 2019.

[22]Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “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).* Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.

[23]Jesse Dodge, Maarten Sap, Ana Marasović, et al. “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.* Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1286–1305.

[24]Kasper Drawzeski, Andrea Galassi, Agnieszka Jablonowska, et al. “A Corpus for Multilingual Analysis of Online Terms of Service”. In: *Proceedings of the Natural Legal Language Processing Workshop 2021.* Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1–8.

[25]Zhangyin Feng, Daya Guo, Duyu Tang, et al. “CodeBERT: A Pre-Trained Model for Programming and Natural Languages”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020.* Online: Association for Computational Linguistics, Nov. 2020, pp. 1536–1547.

[26]Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. *Larger-Scale Transformers for Multilingual Masked Language Modeling.* 2021.

[27]Yu Gu, Robert Tinn, Hao Cheng, et al. “Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing”. In: *ACM Transactions on Computing for Healthcare* 3.1 (Jan. 2022), pp. 1–23.

[28]Daya Guo, Shuo Ren, Shuai Lu, et al. “GraphCode{BERT}: Pre-training Code Representations with Data Flow”. In: *International Conference on Learning Representations.* 2021.

[29]Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, et al. “Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Online: Association for Computational Linguistics, July 2020, pp. 8342–8360.

[30]Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 770–778.

[31]Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention.* 2020.

[32]Peter Henderson, Mark S. Krass, Lucia Zheng, et al. *Pile of Law: Learning Responsible Data Filtering from the Law and a 256GB Open-Source Legal Dataset.* 2022.

[33]Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs).* 2016.

[34]Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, et al. *Training Compute-Optimal Large Language Models.* 2022.

[35]Chalkidis Ilias. *Deep Neural Networks for Information Mining from Legal Texts*. 2021. URL: `http://nlp.cs.aueb.gr/theses/halkidis_phd_thesis.pdf`.

[36]Zihang Jiang, Weihao Yu, Daquan Zhou, et al. *ConvBERT: Improving BERT with Span-based Dynamic Convolution*. 2020.

[37]Xiaoqi Jiao, Yichun Yin, Lifeng Shang, et al. *TinyBERT: Distilling BERT for Natural Language Understanding*. 2019.

[38]Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. *AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing*. 2021.

[39]Yuta Koreeda and Christopher Manning. "ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1907–1919.

[40]Kundan Krishna, Saurabh Garg, Jeffrey P. Bigham, and Zachary C. Lipton. *Downstream Datasets Make Surprisingly Good Pretraining Corpora*. 2022.

[41]Guillaume Lample and Alexis Conneau. *Cross-lingual Language Model Pretraining*. 2019.

[42]Zhenzhong Lan, Mingda Chen, Sebastian Goodman, et al. *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. 2019.

[43]Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* (Sept. 2019). Ed. by Jonathan Wren.

[44]Mike Lewis, Yinhan Liu, Naman Goyal, et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019.

[45]Xiao Liu, Da Yin, Jingnan Zheng, et al. "OAG-BERT: Towards a Unified Backbone Language Model for Academic Knowledge Services". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, Aug. 2022.

[46]Yinhan Liu, Myle Ott, Naman Goyal, et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[47]Antoine Louis. "NetBERT: A Pre-trained Language Representation Model for Computer Networking." PhD thesis. June 2020.

[48]Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, et al. "FiNER: Financial Numeric Entity Recognition for XBRL Tagging". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4419–4431.

[49]Shuai Lu, Daya Guo, Shuo Ren, et al. *CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation*. 2021.

[50]Martin Müller, Marcel Salathé, and Per E Kummervold. *COVID-Twitter-BERT: A Natural Language Processing Model to Analyse COVID-19 Content on Twitter.* 2020.

[51]Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. *BERTweet: A pre-trained language model for English Tweets.* 2020.

[52]Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. *MathBERT: A Pre-Trained Model for Mathematical Formula Understanding.* 2021.

[53]Long Phan, Hieu Tran, Daniel Le, et al. *CoTexT: Multi-task Learning with Code-Text Transformer.* 2021.

[54]A. Radford, Jeffrey Wu, R. Child, et al. "Language Models are Unsupervised Multitask Learners". In: 2019.

[55]Alec Radford and Karthik Narasimhan. *Improving Language Understanding by Generative Pre-Training.* 2018.

[56]Jack W. Rae, Sebastian Borgeaud, Trevor Cai, et al. "Scaling Language Models: Methods, Analysis & Insights from Training Gopher". In: *CoRR* abs/2112.11446 (2021). arXiv: 2112.11446.

[57]Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.* 2019.

[58]Rico Sennrich, Barry Haddow, and Alexandra Birch. "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).* Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725.

[59]Chul Sung, Tejas Dhamecha, Swarnadeep Saha, et al. "Pre-Training BERT on Domain Resources for Short Answer Grading". 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).* Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6071–6075.

[60]Yuqing Tang, Chau Tran, Xian Li, et al. *Multilingual Translation with Extensible Multilingual Pretraining and Finetuning.* 2020.

[61]Jörg Tiedemann and Santhosh Thottingal. "OPUS-MT – Building open translation services for the World". In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation.* Lisboa, Portugal: European Association for Machine Translation, Nov. 2020, pp. 479–480.

[62]Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. "LEDGAR: A Large-Scale Multi-label Corpus for Text Classification of Legal Provisions in Contracts". English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference.* Marseille, France: European Language Resources Association, May 2020, pp. 1235–1241.

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. “Attention is All You Need”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, California, USA, 2017, pp. 6000–6010.

[64] Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. “TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 917–929.

[65] Stratos Xenouleas, Alexia Tsoukara, Giannis Panagiotakis, Ilias Chalkidis, and Ion Androutsopoulos. “Realistic Zero-Shot Cross-Lingual Transfer in Legal Topic Classification”. In: *Proceedings of the 12th EETN Conference on Artificial Intelligence (SETN 2022)*. 2022.

[66] Yi Yang, Mark Christopher Siy UY, and Allen Huang. *FinBERT: A Pretrained Language Model for Financial Communications*. 2020.

[67] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, et al. “Big Bird: Transformers for Longer Sequences”. In: (2020).

[68] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. 2019.

[69] Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. “When does pretraining help?: assessing self-supervised learning for law and the CaseHOLD dataset of 53,000+ legal holdings”. In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law* (2021).

[70] Jie Zhou, Junfeng Tian, Rui Wang, et al. “SentiX: A Sentiment-Aware Pre-Trained Model for Cross-Domain Sentiment Analysis”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 568–579.

List of Acronyms

NLP	Natural Language Processing
AI	Artificial Intelligence
LM	Language Model
RNN	Recurrent Neural Network
BPE	Byte-Pair Encoding
BERT	Bidirectional Encoder Representation from transformers
mBERT	Multi-lingual BERT
RoBERTa	Robustly Optimized BERT pre-training approach
GPT	Generative Pre-trained Transformer
DistilBERT	Distilled-BERT
DeBERTa	Decoding-enhanced BERT with Disentangled Attention
PEGASUS	Pre-training with Extracted Gap-sentences for Abstractive Summarization
BART	Bidirectional Auto-Regressive Transformers
ELECTRA	Efficiently Learning an Encoder that Classifies Token Replacements Accurately
ALBERT	A Lite BERT

ConvBERT	Convolution-BERT
XLM	Cross-lingual Language Modelling
XLM-R	Cross-lingual Language Modelling-RoBERTa
NSP	Next Sentence Prediction
SOP	Sentence Order Prediction
MLM	Masked Language Modeling
CLM	Casual Language Modeling
TLM	Translation Language Modeling
NMT	Natural Machine Translation
ToS	Terms of Service
MAE	Mean Absolute Error
R and D	Research and Development
TN	Τεχνητής Νοημοσύσης
ΓΜ	Γλωσσικών Μοντέλων
ΕΦΓ	Επεξεργασία Φυσικής Γλώσσας

List of Figures

2.1 The transformer-model architecture. The encoder is the left part of the transformer and the decoder is the right part. Source: Vaswani et al. [63] . . 6

2.2 Transformer model with Encoder Stack-Decoder Stack. Source: https://www.baeldung.com/cs/transformer-text-embeddings 7

2.3 Construction of the input of the encoder from embedding vectors and positional encoding vectors. Source: https://www.baeldung.com/cs/transformer-text-embeddings 7

2.4 Illustration of pre-training and fine-tuning of BERT on different tasks. Source: Devlin et al. [22] 8

2.5 Detailed illustration of fine-tuning BERT on different tasks. Source: Devlin et al. [22] 9

2.6 Timeline of some Transformer-based models. There have been two main routes: masked-language models like BERT; and autoregressive models like GPT. Finally, the T5 deserves a special mention thanks to the text-to-text approach it proposes for multi-task learning in NLP. Source: https://factored.ai/transformer-based-language-models/ . . 11

4.1 Train and evaluation loss curves of C-XLMs during pre-training. 22

4.2 MLM performance per language across C-XLM model variants depicted with different coloured webs. 23

5.1 Radar plots with per language performance for the multi-lingual MultiEURLEX and Unfair-ToS datasets for all the versions of C-XLM. 30

List of Tables

2.1 A list of several domain-specific transformer-based language models with their respective papers. Many models in the list were gathered from Kalyan et al. [38] 13

3.1 Model Specifications, Training Tokens processed on pre-training for all variants of our XLM (C-XLM) models and the XLM-R models of Conneau et al. [20] considered as baselines. 17

4.1 Total tokens used per language per pre-training corpus. 20

4.2 MLM Validation Performance Scores (Cross-Entropy Loss, Accuracy) for our C-XLM and XLM-R models. 23

5.1 The table provides information about the 3 publicly available and 2 private datasets. Details like the source, the sub-domain, the number of languages, samples and classes are provided. In addition, the task type is depicted. It should be noted that the task type of UNFAIR-ToS is regression, but the accuracy score is provided through rounded (discrete) scores. 27

5.2 Examples from random samples along with their label for each one of the public datasets. 27

5.3 Optimal Learning Rates per down-stream task across all variants of C-XLM and XLM-R models. Note: CNLI refers to ContractNLI and Obligations to Contract-Obligations tasks. 28

5.4 Overall results our fine-tuned C-XLM models and the baseline XLM-R moodels across all down-stream tasks. Note: CNLI refers to ContractNLI and Obligations to Contract-Obligations tasks. 29

5.5 Micro-F1 (μ-F_1) scores for the MultiEURLEX task per language. Results across all variants of our model C-XLM and the baseline XLM-R are reported. 31

5.6 Macro-F1 (m-F_1) scores for the MultiEURLEX task per language. Results across all variants of our model C-XLM and the baseline XLM-R are reported. 31

5.7 Mean Absolute Error (MAE) and Accuracy (Acc.) scores for the UNFAIR-ToS task per language. Results across all variants of our model C-XLM and the baseline XLM-R are reported. 32