ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

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

Bachelor Thesis
in
Computer Science

Exploration of Efficient Transformer Methods for Long Legal Document Processing

Petros Tsotsi

Supervisors: Ion Androutsopoulos, Ilias Chalkidis

July 2022

Petros Tsotsi
Exploration of Efficient Transformer Methods for Long Legal Document Processing
July 2022
Supervisor: Ion Androutsopoulos, Ilias Chalkidis

Athens University of Economics and Business
School of Information Sciences and Technology
Department of Informatics
Natural Language Processing Group
Athens, Greece

Abstract

Long document processing is a very interesting and challenging problem in Natural Language Processing (NLP) with many applications including long document classification, question answering, coreference resolution and long document matching. Many deep learning approaches have been proposed to deal with long document tasks, with the most successful ones being the pretrained Transformer models which employ a sparse attention pattern that lets them extend their input limit length. Longformer is a well-known model built especially for processing long sequence tasks which follows that paradigm. Focusing on the same goal, we develop Longformer variant models in order to achieve more efficient processing of long, mostly legal, document processing tasks. Among these variants, we try novel settings of the Longformer's attention pattern as well as starting from models already pretrained on a legal corpus. We experiment with those LexGLUE benchmark tasks that consist of long legal texts, i.e., the SCOTUS task, and the ECtHR Tasks A & B, and we then analyze the results of the experiments.

Περίληψη

Η επεξεργασία μεγάλων κειμένων είναι ένα πολύ ενδιαφέρον και απαιτητικό πρόβλημα στην Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing, NLP) με πολλές εφαρμογές, όπως είναι η κατηγοριοποίηση μεγάλων κειμένων, τα συστήματα ερωταποκρίσεων, η ανάλυση αναφορών μεταξύ κειμένων, καθώς και το ταίριασμα μεγάλων κειμένων. Πολλές προσεγγίσεις βασισμένες στη Βαθιά Μάθηση (Deep Learning) έχουν προταθεί για την επεξεργασία μεγάλων κειμένων, με τις πιο επιτυχημένες εξ αυτών να περιλαμβάνουν προ-εκπαιδευμένα μοντέλα τύπου Transformers τα οποία υλοποιούν έναν «αραιό» μηχανισμό προσοχής (sparse attention pattern) που τους επιτρέπει να επεκτείνουν το μέγιστο μήκος κειμένων που μπορούν να επεξεργαστούν. Ένα πολύ γνωστό μοντέλο που υιοθετεί αυτή την πρακτική και έχει αναπτυχθεί ειδικά με σκοπό την αποδοτική επεξεργασία μεγάλων κειμένων είναι ο Longformer. Διατηρώντας και εμείς τον ίδιο στόχο, στην παρούσα εργασία αναπτύσσουμε μοντέλα – παραλλαγές του Longformer με σκοπό την ακόμη πιο αποδοτική επεξεργασία μεγάλων κειμένων, κυρίως μεγάλων νομικών κειμένων. Μεταξύ των μοντέλων που αναπτύσσουμε, δοκιμάζουμε νέες διαφορετικές ρυθμίσεις του μηχανισμού προσοχής (attention pattern) σε σχέση με εκείνες του Longformer, καθώς επίσης και την ανάπτυξη μοντέλων τύπου Longformer ξεκινώντας από μοντέλα προ-εκπαιδευμένα σε ένα σώμα νομικών κειμένων. Στα πειράματά μας εστιάζουμε στα σύνολα δεδομένων του LexGLUE benchmark που περιλαμβάνουν μεγάλα κείμενα, δηλαδή τα SCOTUS και ECtHR A & B, στα οποία αξιολογούμε και αναλύουμε την απόδοση των μοντέλων μας.

Acknowledgements

I would like to thank everyone that has supported me with materials, advice and resources during the work of this thesis. First of all, I would like to sincerely thank my supervisor and professor Ion Androutsopoulos for his continuous support and valuable help and also for giving me the opportunity to work with him on such an interesting research field before ending my Bachelor studies. Next, I would like to thank my other supervisor Ilias Chalkidis for his useful suggestions and ideas regarding the content of this thesis and legal NLP research in general, as well as for his help with technical issues on the hardware demanding experiments that I had to run. Lots of thanks also go to Dimitris Mamakas for the great cooperation we had during the development of our thesis projects under the supervision of Ion and Ilias. Last but not least, many thanks go to my family and friends for their endless support and encouragement during my studies and throughout the whole work of this current thesis.

Contents

Abstract v

Acknowledgements vii

1 Introduction 1

2 Background and Related Work 3
- 2.1 Background . . . 3
- 2.2 Related Work . . . 3
  - 2.2.1 Longformer . . . 3
  - 2.2.2 ETC . . . 5
  - 2.2.3 BigBird . . . 6
  - 2.2.4 Hierarchical Approach - SMITH . . . 8
  - 2.2.5 Recurrent models - ERNIE-DOC . . . 9

3 Implemented Methods 11
- 3.1 Longformer-based models . . . 11
  - 3.1.1 Longformer Extended . . . 11
  - 3.1.2 Longformer Extra Global . . . 12
- 3.2 Legal-pretrained Longformer models . . . 12
  - 3.2.1 Legal Longformer . . . 12
  - 3.2.2 Legal Longformer Extended . . . 13
  - 3.2.3 Legal Longformer Extra Global . . . 13

4 Experiments 17
- 4.1 Datasets . . . 17
  - 4.1.1 LexGLUE Benchmark . . . 17
- 4.2 Evaluation metrics . . . 21
- 4.3 Experimental Results . . . 22
  - 4.3.1 SCOTUS Task, Results & Discussion . . . 23
  - 4.3.2 ECtHR Task A, Results & Discussion . . . 27
  - 4.3.3 ECtHR Task B, Results & Discussion . . . 31

5 Conclusions and Future Work 35
- 5.1 Conclusions . . . 35

5.2 Future Work 35

Bibliography **37**

List of Acronyms **38**

List of Figures **40**

List of Tables **42**

1 Introduction

Natural Language Processing (NLP) is a research field of Computer Science and more specifically of Artificial Intelligence (AI), which develops methods of automatic understanding and generation of human text and speech. Current research efforts in the NLP field mostly focus on machine learning methods and, in particular, deep learning methods which are achieving State of the Art (SoTA) results on numerous NLP tasks from document classification, Name Entity Recognition (NER) tasks, Part of Speech Tagging (POS) to Question Answering (QA) systems and text generation. The most successful deep learning methods in NLP are Transformer models which follow the pretraining-finetuning paradigm, with the example of BERT model [Dev+18] being the most famous. The success of these models comes mainly from the full self-attention mechanism [Vas+17] that they use in order to capture useful information across the entire input sequence that is being processed. However, this attention pattern of the standard Transformer models has a quadratic complexity with respect to the input text length, making it impossible to cope with long document tasks.

Long document processing is a new challenge in NLP with many applications, including document classification, QA tasks , text summarization and also legal dataset tasks which in general consist of very long texts. The effective treatment of this new challenge has been in the focus of many research efforts of the NLP research field in the latest years and thus several approaches have been proposed. The most well-known ones include sparse attention approaches, hierarchical models and recurrent models, which we discuss in Chapter 2 (Related Work) of this thesis. The model that inspired us the most and we focus on, in this present thesis is the Longformer model [BPC20], a sparse attention approach model implemented with focus on the effective long document processing. Longformer proposes a new self -attention component, which scales linearly with the input sequence length, instead of the full self-attention pattern that standard Transformer models use. Longformer's attention pattern consists of a sliding local window whose purpose is to help each input token capture its local context, and a global attention mask which aims, through a symmetric operation on some preselected tokens, to capture a holistic context of the whole input.

Based on these interesting features of Longformer's attention pattern that aim to achieve a successful and effective processing of long document tasks, we implemented new Transformers methods starting from the Longformer model approach. In particular, we propose new Longformer based methods in which we applied novel changes that can hopefully lead to better and more effective processing of long document tasks and especially long legal document tasks. These modifications applied, refer to Longformer's architecture (extension of its input limit length), attention pattern (changing the way global mask attention is applied) and also initialization from domain specific datasets (warm start from legally pretrained models). To evaluate the performance of our proposed methods, we focused on the LexGLUE benchmark dataset [Cha+22], a suite of legal NLP tasks, and especially on three of its tasks which consist of long document datasets. After conducting the experiments of our implemented methods on these long legal documents tasks of LexGLUE, we noticed some interesting results. Firstly, we observe that our methods which are warm started from the legally pretrained checkpoint of Legal BERT [Cha+20], achieve in general much better scores than those simply initialized from Longformer. Moreover, we notice an unstable performance of our ablations across the tasks we evaluated mainly because of the non-even distribution of the dataset instances across the available labels.

Thesis Structure

The rest of this thesis is organized as follows:

Chapter 2

Chapter 2 refers to the background of this thesis and discusses the work that has been done in the NLP research field regarding long document processing, including well known Transformer models such as Longformer, ETC, BigBird, SMITH and Ernie-Doc.

Chapter 3

Chapter 3 describes the models that were implemented during the development of this thesis and were used in our experiments.

Chapter 4

Chapter 4 deals with the datasets and their corresponding tasks that will be used in order to evaluate the performance of our implemented models and presents the results of the experiments that we performed during this thesis along with details and useful statistics of our experimentations.

Chapter 5

Chapter 5 summarizes the work of the thesis and proposes future further work.

2 Background and Related Work

2.1 Background

Transformer based models are currently achieving state of the art results in numerous NLP tasks such as text classification, question answering, text summarization etc. A main part of this success is the full self-attention mechanism, that most Transformer models use, in order to capture useful information across the input sequence.

Although this attention mechanism seems to achieve great results, it brings some significant limitations to the models it's applied. The most important limitation is the memory and computational complexity which scales quadratically with the input sequence length. This is mostly due to the need of each input token to access every other token of the input text in order to compute the attention score for each one and finally compute its own token representation. This limitation makes it difficult for Transformer models to process long input texts like in many NLP and legal NLP tasks. (That's why most Transformer based models like BERT [Dev+18], RoBERTa [Liu+19] and others, using this attention mechanism, have an input limit of 512 tokens).

Many models have been proposed to tackle this issue following different approaches such as Sparse Attention Approach, Hierarchical Approach and Recurrent Models. We are going to discuss some of the most important and well-known models following these approaches including Longformer [BPC20], ETC [Ain+20], BigBird [Zah+20] , SMITH [Yan+20], Ernie-Doc [Din+20].

2.2 Related Work

2.2.1 Longformer

The Longformer model [BPC20] implements a sparse attention mechanism instead of the full self-attention pattern used in the original Transformer models. In particular, it proposes a new self-attention component that scales linearly with the input sequence length and thus it can process tasks with long input text. This attention pattern consists of two main attention parts: a local sliding window and a global attention part.

The local sliding window's main goal is to help each token capture its local context. With local sliding window, each token attends to a specified number of neighboring tokens that surround it. For example, given a local window of size w, each token must attend to w/2 tokens on each side, and not to all input tokens as in the full self-attention mechanism (see Figure 2.1b below). As a result , besides focusing on local context, this local window design has a linear complexity $O(n \times w)$ with the input length n , with the condition that window size w is independent of the input length n.

On the other hand, Global attention aims to learn a representation of the whole input sequence through a symmetric operation , using some preselected input tokens. Specifically, each of these global tokens attend all input tokens and every input token attend to these global tokens. The global attention mechanism is task-oriented, meaning that it is not used or learned during pretraining, but global tokens are placed at preselected positions according to the specific downstream task. For example, in classification tasks only the CLS token acts as a global attention token, while in QA tasks the question tokens are used as global tokens. Since the number of global attention tokens is small and independent of the sequence length, the Longformer's attention complexity combing local window and global attention remains linear. Thus, it allows Longformer to increase the input token limitation from 512 to 4096 tokens long (8 times longer).

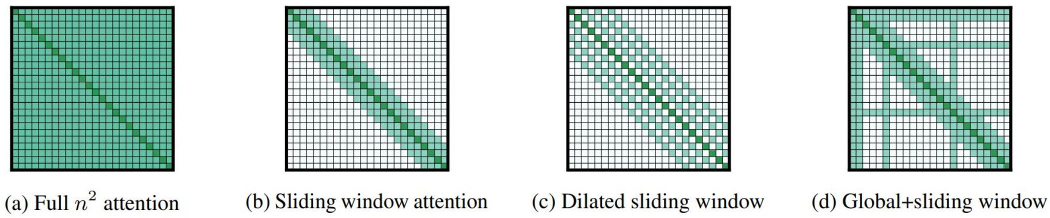


Fig. 2.1: Illustration of the full n^2 attention pattern in comparison with the sliding window attention, the dilated window attention (a variant of the sliding window) and the combination of the global attention with the sliding window as implemented in Longformer.
Figure taken from [BPC20].

Following the same practice as basic Transformer models, Longformer is firstly pretrained and then finetuned for different NLP tasks. However, Longformer is not pretrained from scratch. It continues pretraining from an already pretrained RoBERTa checkpoint [Liu+19] and makes some more pretraining gradient updates to help the model adapt its new sparse attention pattern. As for the positional embeddings, Longformer needs to extend them up to 4096 to support inputs up to 4096 tokens long . Instead of randomly initializing the extra positional embeddings, Longformer uses the 512 pretrained positional embeddings of RoBERTa and copies them multiple times to support 4096 positional embeddings and preserve the local structure of nearby tokens.

2.2.2 ETC

ETC [Ain+20] is another well-known Transformer model, attempting to address the main limitations of the original Transformer models, that include scaling to long inputs and encoding structured inputs. The basic principles of ETC's architecture involve a global-local attention mechanism and a new pretraining objective called Contrastive Predictive Coding (CPC).

ETC's global-local attention mechanism (see Figure 2.2) firstly divides the input sequence into two different inputs, the long input which typically is the entire input sequence and the global input that consists of a small number of auxiliary tokens (independent of input length) each (usually) representing a different segment of the input. Then, four separate attention pieces are computed including long-to-long, long-to-global, global-to-global, and global-to-long attention.

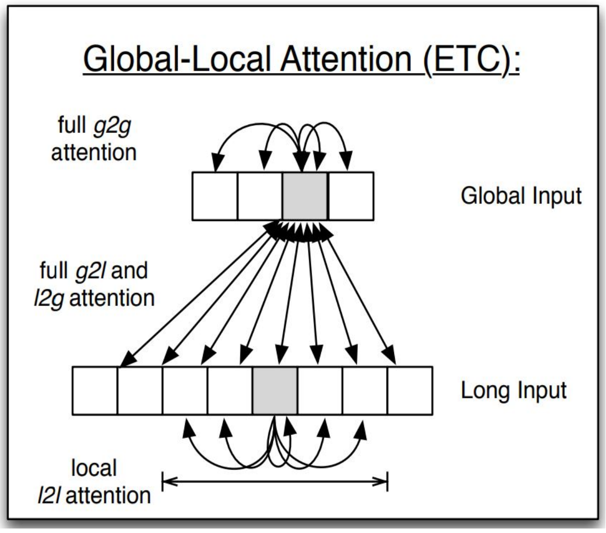


Fig. 2.2: Illustration of the global-local attention mechanism that ETC implements. Figure taken from [Ain+20].

The most computationally intensive piece, long-to-long attention, restricts each token to attend a significant number of nearby tokens in a fixed radius r , rather than attending to every other input token. This idea is very similar to the local sliding window of Longformer's attention. The other three pieces, however, have unrestricted access to compute attention scores. For example, the long-to-global attention allows each input token to attend all global tokens . In that way, this attention design of ETC allows input tokens to transfer useful information to one another through the global tokens despite the long-to-long attention restriction. At the same time the computational complexity is linear with the input sequence length. Considering an input sequence of length n and a global input of length n_g ($n_g \ll n$) , the four attention pieces of ETC will make the following load of computations :

- long-to-long piece : $n \cdot (2r + 1)$,
- long-to-global piece : $n \cdot n_g$,
- global-to-global piece : $n_g \cdot n_g$,
- global-to-long piece : $n_g \cdot n$.

All 4 attention pieces are equal to: $n_g \cdot (n_g + n) + n \cdot (n_g + 2r + 1)$. The final computational complexity of ETC is $O(n_g \cdot (n_g + n) + n \cdot (n_g + 2r + 1))$ where we observe it is linear with the input sequence length n, assuming that n_g « n.

Regarding pretraining, ETC uses a MLM (Masked Language Model) task similar to that of original Transformers with some small modifications. While in the standard MLM task, each token is masked (with some low possibility) and during pretraining the model 'tries' to predict the missing token, ETC uses MLM with a whole word masking, meaning that if a token is masked then all sub-word tokens of the same word will be masked. However, apart from MLM task, ETC proposes a new pretraining objective named CPC aiming to help the model better understand structured inputs. In particular, the CPC pretraining task is like a MLM task at sentence level. During pretraining, some randomly selected sentences are masked (meaning all tokens of these sentences) and the goal of CPC is to predict the hidden representation of these tokens using the global summary tokens of the global input (they are not masked) corresponding to these sentences.

Following the same practice as Longformer, ETC is not pretrained from scratch but is initialized from already pretrained RoBERTa checkpoint (warm start). Subsequently, it makes some more pretraining updates using MLM and the new CPC pretraining objective, to help the model adapt to its new global-local attention mechanism. Moreover, it's worth pointing out that ETC, in contrast with Longformer, pre-trains its global tokens during the pretraining phase by using the CPC pretraining task as described above. On the other hand, in the case of Longformer, the global tokens are not pre-trained, they are only finetuned.

2.2.3 BigBird

Very similarly with Longformer and ETC, BigBird [Zah+20] implements a sparse attention approach aiming to deal with the challenge of processing long input sequences that original Transformers are struggling with, due to the quadratic complexity of their full self-attention component. Big Bird's attention pattern consists of :

a) A local window attention part (see Figure 2.3b), where each token attends to a set of w nearby tokens , w/2 on each side. In that way, BigBird preserves the importance of local context while keeping the attention complexity linear. (w « n , w independent of input length n).

b) A global attention part with a set of g global tokens (g « n) attending all input sequence tokens and attended by all input tokens, through a symmetric operation (see Figure 2.3c). Global tokens can be initialized in two different ways . With ITC configuration, some preselected tokens of the input sequence act as global tokens that attend across the input using this symmetric operation. On the other hand, with ETC configuration, which seems to achieve better results than ITC, extra tokens are added, for example CLS token, to play the role of global tokens .

c) A random attention part where all tokens attend to a set of r random tokens across the input (r independent of input length). (see Figure 2.3a)

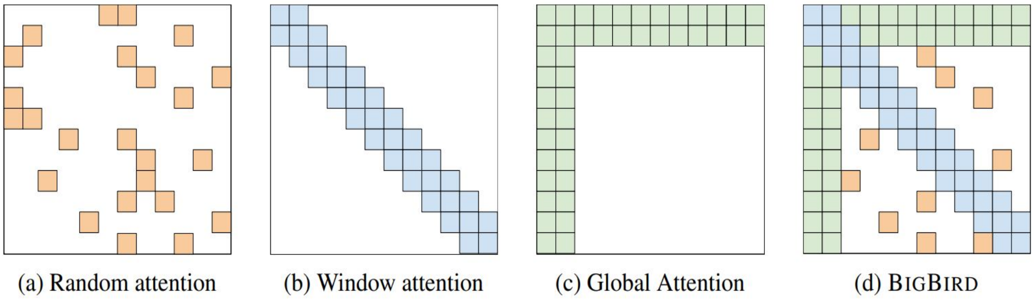


Fig. 2.3: An illustration of the three main components of BigBird's attention pattern. (a) shows the Random attention part, (b) shows the Window attention component, (c) shows the global attention part and (d) depicts the final combination of these 3 components that result to the final BigBird attention pattern.
Figure taken from [Zah+20].

Taking into account these three attention components of BigBird we observe that BigBird's attention complexity is reduced to linear with the number of input tokens. This is easily understood from the fact that each attention component is independent of the input sequence length n. Local sliding window restricts each token to compute attention scores in a specific window w (independent of n), global attention consists of a small number of global tokens (independent of n) attending all tokens with a symmetric operation and random attention requires each token to attend to a set of r random tokens (independent of n).

BigBird, following the same practice as Longformer and ETC, is pretrained, warm starting from a RoBERTa released checkpoint, and subsequently makes extra pretraining steps using the MLM objective. As also stated before, BigBird is initialized in two different configurations, as for the global attention setting, including ITC and ETC. Through ablations and experiments, the developers of Big Bird [Zah+20] showed that the ETC configuration achieves in general better results.

2.2.4 Hierarchical Approach - SMITH

A completely different approach to cope with long document representation learning and matching is the Hierarchical Transformer Encoder Architecture. Several Transformer models have been proposed following this design approach including MatchBERT [Yan+20], Hierarchical Legal Bert [Cha+20] and SMITH [Yan+20].

SMITH (Siamese Multi-depth Transformer-based Hierarchical) model adopts a 2-level encoder architecture focusing on modeling long and structured text inputs to learn final document representations (see Figure 2.4). The main idea, firstly, is to split the input document into sentence blocks of predefined length using a proposed greedy sentence filling algorithm. After the document splitting, the first level of SMITH's encoders, called Sentence level encoders, try to learn contextual representations of every input token in the sentence blocks. Then, each sentence block will be represented by the contextual representation of its first token following the same practice as BERT [Dev+18]. Next, given the representations of each block, the Document level encoders (second level of the encoders architecture) learn contextual representation for each sentence block and finally compute the document representation as the first sentence block representation.

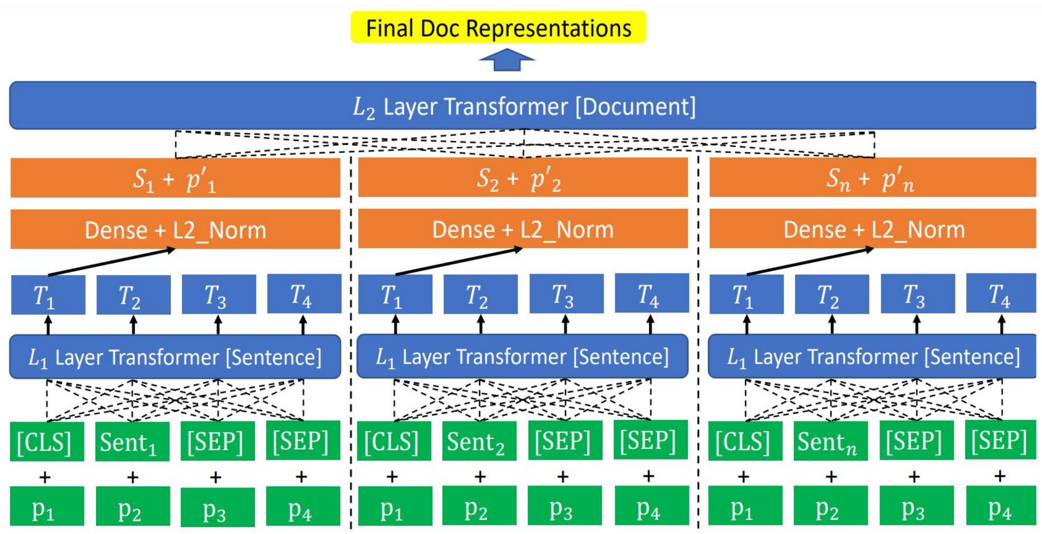


Fig. 2.4: Visualization of the 2-level SMITH hierarchical encoder architecture. Figure taken from [Yan+20]

This architecture design decreases the quadratic computational and memory complexity of basic Transformer models as it replaces the full self-attention pattern of every token attending across all input tokens, to every token attending only the tokens of the same sentence block. As a result, the input document limitation of SMITH is increased at 2048 tokens. Moreover, this design along with the Masked Sentence Prediction pretraining objective allows SMITH to exploit the structure of documents (e.g., passages, paragraphs) to learn context-aware representations of increasingly larger parts of the document structure.

Like all Transformer models, SMITH firstly gets pretrained on large general corpora and then finetuned on downstream tasks. SMITH's pretraining, besides the well-known MLM task, includes the Masked Sentence Block Prediction task. According to this task, some randomly selected sentence block representations are masked and replaced with random weights. During pretraining, all the masked blocks are added in a block pool and the document level Transformers of SMITH produce contextual representations for all the sentence blocks (including the masked ones). Then, using the sentence block representations of the masked blocks, the model will try to predict the true masked sentence block from the these contained in the block pool. Through this objective, SMITH aims to get a better understanding of the sentence block relationships constituting a long document.

2.2.5 Recurrent models - ERNIE-DOC

As stated before, Transformer models cannot cope with processing long documents due to the full self-attention component with the quadratic time and computational complexity. Previously discussed models address this limitation by applying a sparse attention mechanism as a drop-in replacement of the full attention component (ETC , BigBird, Longformer) while others implement a hierarchical approach like SMITH. An additional remarkable approach on long document-level processing includes recurrent models. The main feature that distinguishes them among others, is that they (the recurrent models) allow each document segment to use the contextual information of previous segments to compute the current segment's representation. A model worth discussing, which follows this approach and also introduces several interesting novel techniques, is ERNIE-DOC [Din+20].

ERNIE-DOC (Retrospective Long-Document Modeling Transformer) is a recurrent, but still Transformer-based model aiming to deal with processing long documents. It adapts two innovative techniques: Retrospective feed mechanism and Enhanced recurrent mechanism (see Figure 2.5 that follows).

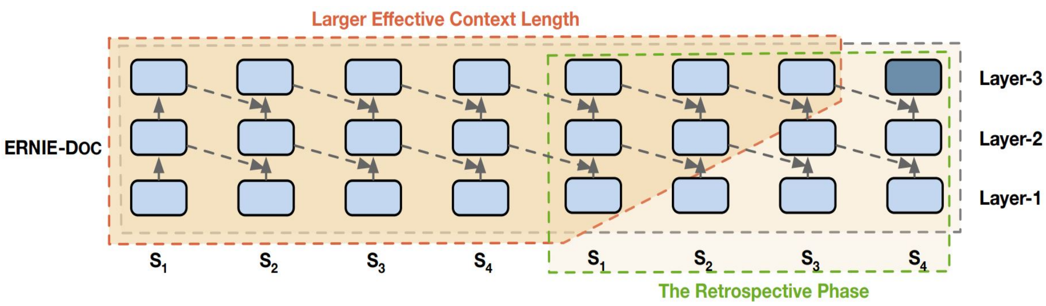


Fig. 2.5: Visualization of the general architectural organization of Ernie-DOC model including the Enhanced recurrent mechanism and the Retrospective Feed phase.
Figure taken from [Din+20]

Most existing approaches such as Sparse Transformers and simple recurrent models face the problem of unavailability of the contextual information of the whole input document while computing the representations of each segment during the training. The retrospective feed mechanism is employed by ERNIE-DOC to address this limitation. The main idea of this mechanism is that the document segments are fed twice as input, in a first phase called 'skimming phase' and a second time called 'retrospective phase'. In the skimming phase, a recurrence mechanism is employed in order to encode the hidden states for each segment, while in the retrospective phase the text segments reuse the cached hidden states from the previous phase which contain the contextual representation of the whole input document. In this way, during the retrospective phase each segment will be able to attend to useful information coming from the entire document learned from the skimming phase with no information loss. It is similar with a human reading a text, where firstly we usually make a skimming parse of text and subsequently, we make a second more concentrate reading of the text.

The enhanced recurrent mechanism also plays a key role in ERNIE-DOC's functionality. In Recurrent models, as stated before, each segment uses the hidden states of previous segments to compute its own segment representation (one layer downwards recurrence). However, with this practice, the maximum context length of each segment is limited by the number of layers, since the largest effective context length grows linearly with the number of layers. For example, if a simple recurrence mechanism was applied on the Figure's 2.5 model, the S4 upper layer representation could only fuse the contextual representation of the two previous segments. To cope with that limitation, enhanced recurrent mechanism replaces the 'one-layer-downwards-recurrence' with a same layer recurrence , where every document segment now can attend to previous higher-level representations. As we can see from the Figure 2.5, every segment in the retrospective phase (in the green dotted box) has now a bidirectional information of the entire input, taking advantage of the the skimming phase and the enhanced recurrent mechanism that allows each one to attend to all input segments with no layer limitations.

Regarding pretraining, besides MLM (Masked Language Model) objective , ERNIE-DOC uses a novel pretraining objective called Segment-Reordering Objective. According to Segment reordering objective, pretraining documents are firstly split into chunks and then randomly shuffled. After this, the shuffled document is processed in split segments, and the goal of the task is predicting the correct order of the permuted chunks. With this objective, ERNIE-DOC learns to model relations between segments of the input documents.

3 Implemented Methods

This section's purpose is to describe in detail the methods that were implemented during the work of this thesis. As also stated in Chapter 1, our main focus is the deployment of Transformer based methods capable of handling long document tasks in an efficient way. This problem still remains challenging, since current SoTA Transformers, like BERT [Dev+18], have some significant limitations that make it impossible to cope with tasks including long input sequences. The most important limitation is their self-attention operation which scales quadratically with the input sequence length.

Many approaches and models have been proposed to address that problem, most of which were discussed in Chapter 2. Among them, Longformer [BPC20] was the one that inspired us the most to develop our new proposed methods. Longformer's basic approach is to replace the full self-attention pattern of standard Transformers with a new sparse attention mechanism that scales linearly with the sequence length. This allows Longformer to extend its input limit from 512 to 4096 tokens long and thus to be able to process long sequences.

3.1 Longformer-based models

We start from the first two Longformer based variants that we implemented, in which we apply some new ideas we call "Extended" setting and "Extra Global" setting, both aiming to deal with the effective processing of long document tasks. We refer to the Longformer Extended model, and Longformer Extra Global model.

3.1.1 Longformer Extended

Despite the remarkable success of Longformer on long document tasks including text classification, QA, text summarization, etc., we observe that there are still some demanding NLP tasks that require even longer input limit (than the 4096 limit of Longformer) in order to be fully processed without any partition or truncation of their input text. That restriction was what inspired us to implement what we call Longformer-Extended.

Longformer-Extended is a Longformer variant we developed that will be able to process effectively even longer inputs than the 4096-limit restriction of Longformer. The main idea of Longformer-Extended is to extend the input sequence limit length from 4096 up to 8192 tokens, in other words doubling the input length limit of the standard Longformer. To achieve that, we firstly double the positional embeddings (from 4096 to 8192) and then we clone the weighs of the existing 4096 positional embeddings of Longformer to initialize the embeddings of the new 4096 positions. In that way we preserve the local structure of nearby tokens instead of randomly initializing the weights of the new positional embeddings. Moreover, we decrease the size of the local sliding window from 512 (in standard Longformer's configuration) to 128 tokens trying to balance the computation load. Also, it is remarkable that Longformer-Extended is not more pretrained than the standard Longformer. Simply, starting from standard Longformer, we extend the limit to 8192 tokens through copying the existing positional embedding to the new ones and then allow backpropagation during finetuning to modify all the positional embeddings.

3.1.2 Longformer Extra Global

Continuing from Longformer-Extended we implement another Longformer -based model that we call Longformer-Extra-Global. Specifically, starting from the Longformer-Extended variant model described above, with the 8192-input limit length and the smaller local windows of 128 tokens length, we apply a new global mask setting. According to this new setting, we set periodic global tokens at the start of each paragraph instead of using fewer global tokens at preselected positions. For example, the Longformer standard model in classification tasks, sets the CLS token at the beginning of the text as a global token.

With this alternative setting and design, we hope that Longformer-Extra-Global will be able to capture better the input, because of these periodic global tokens and their symmetric operation. We also expect it to cope better with long structured inputs, because of the global tokens being selected across each paragraph.

3.2 Legal-pretrained Longformer models

The following subsections present the Longformer based variants that were implemented during this thesis, which unlike the above ones, are warm started from the legally pre-trained model checkpoint, Legal BERT [Cha+20]. These include Legal Longformer, Legal Longformer Extended and Legal Longformer Extra Global.

3.2.1 Legal Longformer

As also discussed in Chapter 2, Longformer is not initialized or pretrained from scratch, but it continues pretraining from an already pretrained RoBERTa [Liu+19] checkpoint making only some extra gradient updates in order to adapt to its new positional embeddings (RoBERTa has 512 positional embeddings, and they are copied 8 times until the 4096 Longformer limit) and its new attention pattern.

Following the same practice, we implement Legal Longformer, a Longformer variant that is initialized from a Legal BERT pretrained checkpoint in the same way as standard Longformer was initialized from RoBERTa. Specifically, we warm-start from a Legal BERT checkpoint, a BERT type Transformer model that is pretrained in legal corpora, and then extend its 512 positional embeddings up to 4096 just like standard Longformer does when warm-starting from RoBERTa. However, unlike standard Longformer, when implementing Legal Longformer we make no further pretraining after warm-starting from Legal BERT, we just warm-start and fine-tune.

In future work, we plan to make some extra minimal continued pretraining for about 20-50k steps in order to help the model to tune the additional positional embeddings.

3.2.2 Legal Longformer Extended

Legal Longformer Extended is a new legally-pretrained Longformer based variant which combines both settings of Legal Longformer model and Longformer Extended, as they are described above in their corresponding subsections.

In particular, the main idea is to start from the setting of Legal Longformer, that is warm started from a Legal BERT checkpoint pretrained in legal corpus, and shares the same setting with standard Longformer model (Max input length = 4096, Attention window size = 512). In addition to this, we apply the "Extended" setting that was used for the implementation of Longformer Extended model, which means that we extend Legal Longformer to 8192 input limit length while decreasing the attention window size to 128.
In this way, we developed what we call Legal Longformer Extended, a model that can be seen from two different points of view. From one point as an extension over Legal Longformer model that doubles its input text length capabilities and from another point as a specialized legal Longformer Extended variant that is warm started from a legal pretrained model - Legal BERT.

3.2.3 Legal Longformer Extra Global

Legal Longformer Extra Global model is another variant we developed following the idea of Longformer based models that are legally pretrained. What's new about this model is the combination of the Legal Longformer Extended setting, discussed in the previous subsection, with the Extra Global setting of subsection 3.1.2.

The Legal Longformer Extended is a model warm started from Legal BERT and extended to an 8192 input max limit length. Based on this model setting, we apply a new attention pattern which replaces its global attention setting with the “Extra Global” approach. “Extra Global” defines a periodic distribution of the global tokens across the input texts being processed during the finetuning phase instead of fewer global tokens (e.g., the CLS tokens at the beginning of the input text).
As in Legal Longformer Extended, this new variant can be seen from two points of view. From one point, as an enriched extension over Legal Longformer Extended as we apply to it our new proposed global attention setting, and from another point as a Longformer Extra Global variant which is warm started from a legally pretrained checkpoint.

Model	**Max Input Length**	**Window Size**	**Global Tokens Positions**	**Pretraining Corpora**	**Warm-started from**
Longformer-Standard [BPC20]	4.096	512	CLS Token	General Corpus	RoBERTa
Longformer Extended	8.192	128	CLS Token	General Corpus	RoBERTa
Longformer Extra Global	8.192	128	CLS+SEP Tokens	General Corpus	RoBERTa
Legal Longformer	4.096	512	CLS Token	Legal Corpus	Legal BERT
Legal Longformer Extended	8.192	128	CLS Token	Legal Corpus	Legal BERT
Legal Longformer Extra Global	8.192	128	CLS+SEP Tokens	Legal Corpus	Legal BERT

Tab. 3.1: A summary of the implemented methods presented in this section, along with their main features. Longformer-Standard refers to the original Longformer model presented in the corresponding paper [BPC20].

4 Experiments

In this chapter, the main point of discussion will be the experiments that were performed within the study of this thesis. First of all, we refer to the datasets and the evaluation measures that we used in order to evaluate the performance of our proposed models. Next, we present all the experiments that were executed which include applying the proposed models of Chapter 3 to the tasks and datasets that we discuss in the following section. Moreover, we focus on analyzing the results of the experiments held.

4.1 Datasets

In this section we are going to discuss the datasets we will use in order to evaluate the performance of our implemented methods that were discussed in the previous Chapter 3 (see Table 3.1). Our main focus will be on large text datasets that current state of the art Transformer based models struggle to cope with (mostly because of their input length limitations). With this goal in mind, we chose to experiment with LexGLUE benchmark [Cha+22], a legal NLP set of tasks and especially with the ones with long input texts (SCOTUS , ECtHR Task A , ECtHR Task B).

4.1.1 LexGLUE Benchmark

LexGLUE (Legal General Language Understanding Evaluation) benchmark [Cha+22] is a collection of seven English legal NLP datasets that can be used to evaluate the performance of any proposed NLP method across a variety of legal NLU tasks. LexGLUE is designed to push towards generic-pretrained models that can cope with multiple NLP tasks with some limited extra training (fine tuning) for each one and that approach suits well with our implemented models and goals. We aim to build models that will follow the paradigm of pretraining and finetuning and will be capable to cope efficiently with tasks with long input texts. Towards this direction, LexGLUE can be a great challenge for our implemented methods and especially with its long text tasks such as SCOTUS and ECtHR Tasks A and B that we discuss below. Table 4.4 at the end of this subsection presents a summary of all LexGLUE datasets and their main characteristics.

SCOTUS Task

The SCOTUS task is a single-label, multi-class classification problem where the task, for the evaluated model, is to predict the relevant issue area a court opinion document belongs to (see Table 4.1). These court opinions come from the US Supreme Court (SCOTUS), which is the highest federal court in the US and hears only complex and controversial cases that were not fully or efficiently solved by lower courts. The SCOTUS dataset cases are chronologically split into training set (5.000 cases from 1946 to 1982), development set (1.400 cases from 1982 to 1991) and test set (1.400 cases from 1991 to 2016) and each of these cases is mapped to its relevant issue area, one of the 14 areas available (meaning 14 labels to predict from).

One of the main features of this task, that makes it interesting to deal with, is the text input length of its cases that overcome by far the input token limits of the standard Transformer models (512 tokens limit) and even the limits of models specialized in processing long texts such as Longformer and BigBird (4096 tokens limit).

SCOTUS Text Example	Label
'329 U.S. 167 S.Ct. 691 L.Ed. 3OIL WELL CEMENTING CO.v.WALKER, et al.. 24.Oct. 23, 24, 1946.Nov. 18, 1946..Earl Babcock, of Duncan, Okl. (Harry C. Robb, of Washington, D.C., on the brief), for petitioner.. Harold W. Mattingly, of Los Angeles, Cal., for respondents.. Justice BLACK delivered the opinion of the Court.1P. Walker, owner of Patent No. 2,156,519, and the other respondents, licensees under the patent, brought this suit in a federal district court alleging that petitioner, Halliburton Oil Well Cementing Company, had infringed certain of the claims of the Walker patent. The district court held the claims in issue valid and infringed by Halliburton. The circuit court of appeals affirmed, 9 Cir., 146 F.2d 817, and denied Halliburtoń petition for rehearing. 149 F.2d 896. Petitionerś application to this Court for certiorari urged, among other grounds, that the claims held valid failed to make the ́full, clear, concise, and exact ́description of the alleged invention required by Rev.Stat. § 4888, 35 U.S.C. § 33, 35 U.S.C.A. § 33,1 as that statute was interpreted by us in General Electric Co. v. Wabash Appliance Corporation, 304 U.S. 364, 58 S.Ct. 899, 82 L.Ed. 1402.2 This statutory requirement of distinctness and certainty in claims is important in patent law. We granted certiorari to consider whether it was correctly applied in this case. [...]	7 - Economic Activity

Tab. 4.1: An example of a SCOTUS text case instance together with its ground truth SCOTUS label. The text example is truncated for space purposes.

ECtHR Task A

ECtHR task A is a multi-label, multiclass classification problem also included in the LexGLUE benchmark. Using a dataset that contains cases of allegations (organized in a list of factual paragraphs) that a state has breached human rights of the ECHR (European Convention of Human Rights), the goal of the examined model is to predict the articles of the ECHR that were violated (see Example 4.2). The task dataset includes only 10 ECHR articles , so each case can be mapped to one or more of these 10 articles-labels (multi-label task) or even none of them (+1 pseudo label for no article match).

Following the same practice as with the SCOTUS dataset, ECtHR cases are split chronologically into train, development, and test sets. Specifically, the training set contains 9.000 cases from 2001 to 2016, development set contains 1.000 cases from 2016 to 2017 and the test set consists of 1.000 cases from 2017 to 2019.

ECtHR (A) Text Example	**Label**
11. On the basis of the Land Ownership Act, the applicant entered into restitution agreements with two legal persons (the Hradec Králové State Forest Enterprise and the Líny – Krásná Ves Agricultural Cooperative) on 10 December 1993 and 4 May 1994 respectively. By two decisions of 12 October 1994 the Mladá Boleslav Land Office (pozemkový úřad, "the Land Office") refused to approve the restitution agreements. Referring to section 32(3) of the Land Ownership Act, it found that some of the plots had been assigned to different owners pursuant to the 1948 Act, and that these owners, being natural persons, had proved their property rights by showing their deeds of assignment. The Land Office based its decisions on the following documents: the decision of the former Mladá Boleslav State Notary (státní notářství) of 26 May 1977 on the applicant's father's inheritance, the decision of the former Doksy District National Council of 7 June 1949 on the expropriation of the applicant's father's property, the record of the former Líny Local National Council (místní národní výbor) of 7 November 1949 on the proceedings on the applicant's father's appeal against the expropriation, the decision of the former Liberec Regional National Council (krajský národní výbor) of 29 November 1949 by which the decision of expropriation had been modified, and an extract (výpis) from the land register (pozemková kniha) relating to the Líny and Krásná Ves Districts. [...]	3 - Right to a fair trial

Tab. 4.2: An ECtHR Task A case example (list of factual paragraphs) along with the ECHR Article violated. The case text is truncated for space purposes.

ECtHR Task B

ECtHR Task B is a multi-label, multiclass classification problem like ECtHR Task A. These 2 tasks, also, use exactly the same dataset and cases (cases of allegations that a state has breached human rights of the ECHR). The main difference however between them is that in task B the evaluated model has to predict which of the 10 articles of the ECHR were "allegedly" violated for each case (see Table 4.3), while in task A the challenge was to map each case with the articles that were truly violated.

It can be easily understood that task A is more challenging than task B, since the truly violated articles need a more thorough understanding of the case in order to be predicted than the allegedly violated ones. This can be observed in the next chapter where we report the results on these two tasks for every model we have implemented and where it can be seen that task B scores are generally much better than the ones in task A.

ECtHR (B) Text Example	Label
'8. The applicant is a British national, born in 1940 and living in London.', "9. The applicant, who has a sociology degree, is a journalist and broadcaster. He has written for many national newspapers and magazines and has made a number of appearances on radio and television. In September 1995 an article was published in Spiked magazine in which the applicant suggested that the athlete Linford Christie used banned performance-enhancing drugs. The article stated, inter alia :"On the basis of circumstantial evidence many believe, but cannot prove that Christie has been taking performance-enhancing drugs ... If he has been outwitting the testers for years, it is extremely unlikely that Christie will be caught in the few months left before his likely retirement from competitive sprinting. Nevertheless, there is no bloody hypodermic needle, and no direct evidence that points the finger at Christie. ...Certainly the ten days between injuring himself in Gothenburg and winning in Zurich would have allowed Christie to recover from a slight hamstring injury and, without fear of a random test, put in seven days intensive training, boosted by banned drugs, and perhaps human growth hormone, that would give him the explosiveness and power to run 10.03 seconds into a headwind. We don't know. ...Christie exhibits a number of other possible effects of these performance-enhancing drugs.[...]	3 - Right to a fair trial 6 - Freedom of Expression

Tab. 4.3: An ECtHR Task B case example instance (list of factual paragraphs) along with the ground-truth ECHR Articles that were allegedly violated. The case text is truncated for space purposes.

Dataset	Task Type	Task Instances	Task Labels	Average Word Length
SCOTUS	Single-Label Classification	5.000/1.400/1.400	14	5.939/8.928/8.819
ECtHR Task A	Multi-Label Classification	9.000/1.000/1.000	10+1	1.620/1.785/1.926
ECtHR Task B	Multi-Label Classification	9.000/1.000/1.000	10+1	1.620/1.785/1.926
EUR-LEX	Multi-Label Classification	55.000/5.000/5.000	100	1.134/1.313/1.872
LEDGAR	Single-Label Classification	60.000/10.000/10.000	100	114/113/108
UNFAIR-ToS	Multi-Label Classification	5.532/2.275/1.607	8+1	33/34/33
CaseHOLD	Multiple choice QA	45.000/3.900/3.900	-	-/-/-

Tab. 4.4: A summary of the LexGLUE task datasets including their main features and statistics. (Task Instances and Average Word Length columns refer to train/development/test subsets of each task respectively).

4.2 Evaluation metrics

Before, presenting and discussing the results of our experiments it's important to define the evaluation metrics that are going to be used in order to measure the performance of our models. The metrics that we apply for the evaluation of our following experiments is F1 score and more specifically its averaging metrics: micro-F1 and macro-F1 scores. F1 score is a very frequently used measure that combines both Precision and Recall metrics in an effective way. It is also known as the harmonic mean of Precision and Recall.

Making an assumption of a 2-label classification problem with one label referred as the "Positive" label and the second one as the "Negative" label, Precision is defined as the measure of how many of the predictions classified as "Positive", by the tested model, are truly positive (see Equation 4.1). On the other hand, Recall measures how many of the truly "Positive" instances are correctly classified as "Positive" by the model we evaluate (see Equation 4.2).

In the following equations, TP (True Positive) refers to the number of the "Positive" instances that are truly classified as "Positive" by the model. FP (False Positive) denotes the number of the "Negative" instances that are wrongly classified as "Positive". FN (False Negative) is the number of the "Positive" instances that wrongly are classified by the model as "Negative".

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

The F1 Score can be defined as in Equations 4.3 or 4.4 following :

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.3}$$

$$F1\ Score = \frac{TP}{TP + \frac{1}{2} * (FP + FN)} \tag{4.4}$$

However, in the above scenario where Precision and Recall metrics were defined, the problem involved only 2 labels. In general, most classification problems, including the ones discussed in section 4.1 that we will experiment, consist of more than 2 labels. In this situation Precision, Recall and F1 score metrics are computed for each label separately. Thus, instead of having all these metrics for each class it's usually preferred to compute an average of them and conclude with a single number of the overall performance of the examined model. The most widely used averaging methods are macro F1 score (m-F1), also known as unweighted mean, and micro F1 score (μ-F1).

Macro F1 (m-F1) is computed as the average (unweighted mean) of all the per-label F1 scores. This means that it treats all labels equally regardless their appearance in the train set or any other factor.

As for the **Micro F1** (μ-F1) score averaging metric, it firstly computes the TP, FP and FN instances of each class separately based on the examined model's predictions and then sums the TP's of all classes , the FP's classes and so on. Finally, it computes the average F1 score using these final TP, FP, FN values as in Formula 4.4.

In the following experimental results, we report both micro-F1 and macro-F1 average scores as well as F1 scores per label for a deeper analysis of the average scores observed.

4.3 Experimental Results

For our experiments, we use the publicly available Hugging Face checkpoints of Longformer [BPC20] and Legal BERT [Cha+20] as well as the Hugging Face Datasets to have access to LexGLUE benchmark datasets. We train our models using the Adam optimizer, with a learning rate of 3e-5 and up to 20 epochs with an early stopping callback setting of 3 epochs patience in the development set. For each experiment (meaning for each task and examined model) we perform five iterations, with different random seeds, and choose that seed with the best performance (macro F1 score) in the development set of the corresponding task.

Then, we report the final test results with that best seed for each experiment in Table 4.5.

Method	SCOTUS μ-F1 / m-F1	ECtHR Task (A) μ-F1 / m-F1	ECtHR Task (B) μ-F1 / m-F1
Longformer - standard	* 73.9 / 64.6	* 71.0 / 63.6	78.3 / 71.1
Longformer Extended	73.7 / 63.6	70.9 / 62.1	79.2 / 73.9
Longformer Extra-Global	73.9 / 66.0	70.8 / 62.3	79.0 / 73.1
Legal Longformer	76.6 / 66.9	**71.7** / 63.6	80.5 / **76.4**
Legal Longformer Extended	**77.5** / **67.3**	71.2 / 64.3	**81.4** / 74.2
Legal Longformer Extra-Global	76.2 / 66.3	71.4 / **68.4**	79.6 / 73.9
Hierarchical Legal BERT	76.4 / 66.5	70.0 / 64.0	80.4 / 74.7

Tab. 4.5: Test results for all implemented models across LexGLUE tasks, with the seed with the best development score.
* In these 2 experiments, there were 2 seeds with the same macro F1 score in the development set and we reported the test results of the seed with higher test scores.

4.3.1 SCOTUS Task, Results & Discussion

During the discussion of the LexGLUE benchmark in Chapter 4.1, we observed that SCOTUS is the dataset with the greatest average word length comparing with the other LexGLUE datasets. According to Table 4.4, the SCOTUS train set has an average of almost 6.000 words while the development and test set have an average of almost 9.000 words. This means that SCOTUS text instances overcome (in average) by far the maximum input length of Transformer based models like BERT, that equals to 512 tokens. In addition, we can also see that even models implemented especially to cope with long document tasks such as Longformer [BPC20], ETC [Ain+20], BigBird [Zah+20] etc. cannot deal with the SCOTUS dataset text instances without any modification or truncation of their text. These models, as discussed in Chapter 2 (Related Work), set a maximum input token length of 4096 for the instances that are going to be processed, which is still lower than the SCOTUS dataset needs. This means that when running the standard Longformer model [BPC20] on the SCOTUS task, inevitably some parts of the text instances are going to be truncated with possible valuable information being cut off. It's easily understood that for the SCOTUS task there is a need for models that can cope with even longer texts, and this need we hope to fulfill with our proposed models of Chapter 3.

Table 4.5 shows the results of running the standard Longformer model [BPC20], as well as our proposed methods discussed in detail in Chapter 3 on the SCOTUS task. What can be first noticed is that the best scores are achieved by the Legal Longformer Extended model. This means that the Extended setting, which doubles the max input limit, together with the warm start from the legally pretrained checkpoint of Legal BERT [Cha+20], outperforms all the other proposed settings applied as well as the Hierarchical Legal BERT architecture used in LexGLUE paper [Cha+22], both in micro F1 and macro F1 metrics.

Regarding the non-legally pretrained models (Longformer Extended and Longformer Extra Global) we notice a small decrease in the scores of the former (0.2% in micro F1 and 1% in macro F1) and an important increase, about 2.5%, in the macro F1 score of the latter, both in comparison to the Standard Longformer scores. A possible explanation of the Longformer Extended slightly decreased scores can be the difficulty of the model to adapt to the doubled positional embeddings that have not been 'learned' during pretraining and the fact that no extra gradient updates have been applied for the weights of the new positional embeddings. On the other hand, the improvement we observe on the Extra Global model (which is a variant continuing from the Extended setting that adds periodic global tokens across the input) can be explained by the periodic global tokens that help the model capture more information across the long input texts of the SCOTUS dataset, which possibly are truncated by the standard Longformer model or not taken fully into account by the Extended setting due to the restricted one global CLS token.

As far as the legally pretrained models are concerned (Legal Longformer, Legal Longformer Extended, Legal Longformer Extra Global), we observe a clear improvement on the scores that they achieve in comparison with the models not pretrained in legal corpora we previously discussed. This can be easily explained by the fact that SCOTUS is a legal dataset task and our models that are warm started from the legal pretrained checkpoint of Legal BERT [Cha+20], are more able and flexible to cope with legal words, expressions, and internal structure of legal texts than those pretrained on a general corpus.

For a more detailed discussion regarding the SCOTUS experimental results, in Figure 4.2 we present the F1 scores achieved per label from all models examined in this task. Firstly, we can observe that the Longformer model as well as our proposed models achieve low F1 scores in Labels 3, 4, 8 and 9. According to Figure 4.1, which depicts the distribution of the SCOTUS dataset instances (from the train and test set) across all labels, we can see an extremely small number of train and test instances that correspond to labels 3,4 and 9. In particular, we found that 193 train instances and 51 test instances correspond to Label 3, 45 train instances and 28 test instances correspond to Label 4 and Label 9 includes 191 train and 83 test instances. From this analysis, what we can conclude is that, in general, labels with a small number of train and test instances seem to be a great challenge for all our tested models, which is intuitively obvious, from the fact that a model that is trained on a very small number of instances of one label is not able to generalize across new examples of this particular label. However, this does not seem to be an absolute rule. As we can also see from Figure 4.1, labels 5 and 10 also have an extremely low number of corresponding examples both in train and test subsets, but all our examined models seem to achieve very high scores on these labels. This can be possibly explained by hypothesizing that these labels may be easier for the models to understand and generalize or they may be semantically close with other labels that our models are exposed to with many examples during training.

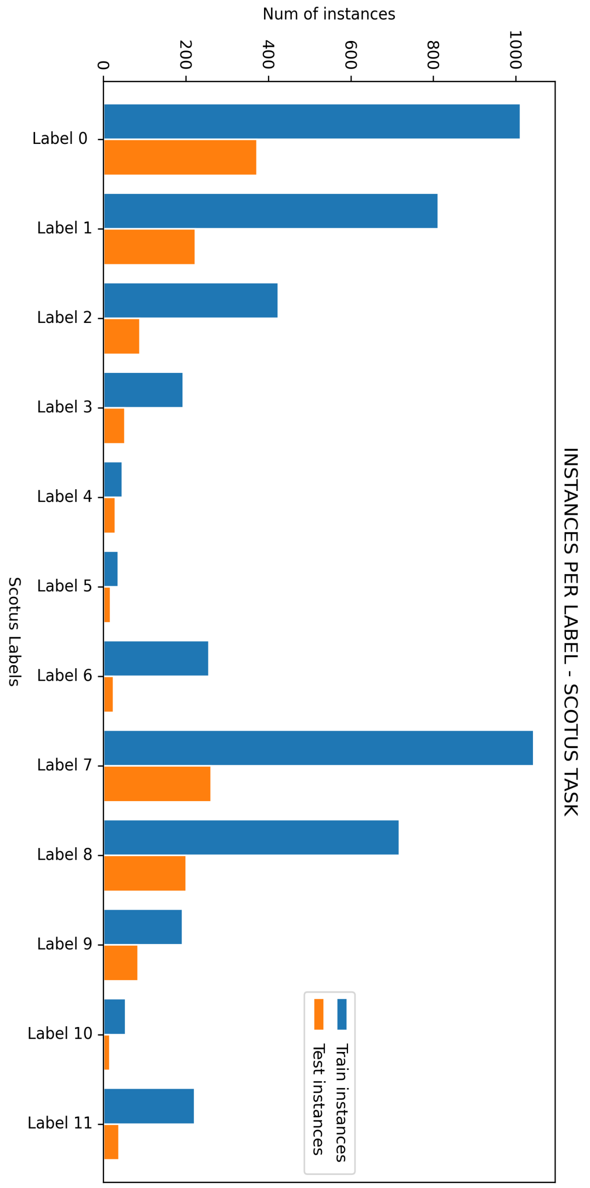
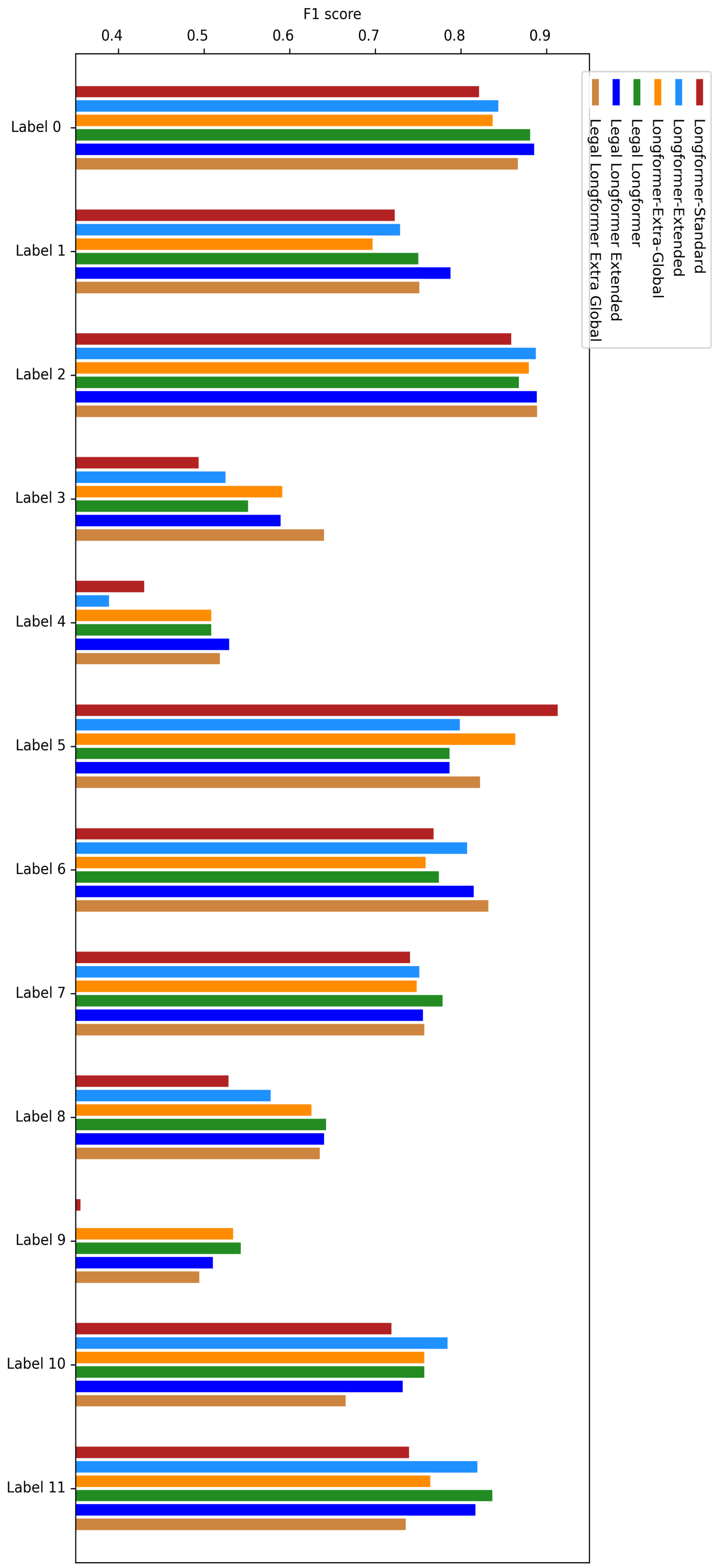


Fig. 4.1: Distribution of train and test instances in SCOTUS dataset across all SCOTUS labels. Note that labels 12 and 13 are not shown due to their minimal or zero appearance in the dataset.

Fig. 4.2: Figure showing the F1 scores achieved by each of our implemented models in SCOTUS task across all SCOTUS labels.



4.3.2 ECtHR Task A, Results & Discussion

The ECtHR dataset, which is the same for both ECtHR tasks A & B, is a long legal document dataset with approximately 1.600, 1.700 and 1.900 average tokens across the train, dev and test subsets respectively, according to Table 4.4. This means that an average text instance of the ECtHR dataset will exceed the limit of 512 input tokens of BERT [Dev+18] or BERT type models, while in most cases, it will be normally processed with no truncation by models like Longformer or BigBird which set a greater limit of 4096 tokens.
Having that in mind, we tested all our implemented methods in the ECtHR Task A and present the results in Table 4.5. What we can first see is a difficulty of the non-legally pretrained methods (Longformer Extended, Longformer Extra Global) to achieve better scores comparing with the standard Longformer. An obvious explanation of that can be the average length of the ECtHR texts which in general does not exceed the 4096-limit of Longformer and so there is no actual need to extend that limit even more as the Extended and the Extra Global settings do.

Regarding our implemented models that were warm started from the legally pretrained Legal BERT [Cha+20], we observe a general improvement of the scores, with Legal Longformer obtaining the highest micro F1 score and Legal Longformer Extra Global achieving the highest macro F1 score. As in the SCOTUS task results, these improvements are expected because of the fact that a model that is warm started from Legal BERT, which is pretrained on legal corpora, will be much more able to process efficiently a legal dataset task than those not pretrained in legal texts.

For a deeper analysis of the results of ECtHR Task A, in Figure 4.4 we report the F1 scores achieved per label by each of our proposed models together with Longformer standard model and Hierarchical Legal BERT [Cha+22]. Also note that the last label named “No Label” is an extra pseudo-label we added to map those instances that do not match any of the 10 ECtHR labels of the ECtHR dataset. Taking a careful look of Figure 4.4 which shows the F1 scores per label in ECtHR Task A, we see that most tested models struggle to achieve high scores on labels 5, 8 and “No Label” in comparison to the other labels, and as a result the average scores achieved are decreased mostly because of these 3 labels.
This is an important observation, because finding the reason of why there is a difficulty on these particular labels can result not only in improvement in the scores of these labels but also in improvement of the average scores achieved. Trying to explain why these particular labels seem so difficult for our models to cope with, we refer again to the train/test instances distribution across all ECtHR labels (see Figure 4.3). We observe that both labels 5 and 8 have an extremely small representation in the train and test datasets. In particular, the examined models are trained only on 41 instances corresponding to label 5 and tested on just 5 examples of the same label. As for label 8, we found 141 train examples and just 16 test instances. As in the SCOTUS task discussion, this small representation of these labels

can be a possible explanation of why most tested models cannot achieve better scores. And we refer to "possible" explanation because we observe that there are also some labels like 6 and 7, which despite being rare in the training set, are more "easily" proceeded and bring high F1 scores.
However, the pseudolabel "No label" which also seems to bring low results across all models we ran, does not have an extremely low instance representation in the ECtHR dataset (914 train examples and 153 test examples) that could explain the low scores, as in label 5 and 8 we discussed previously. In this case, we refer to the special role of that label which, as we discussed before, is a pseudolabel added in order to map the examples that do not match with any other label (ECtHR Tasks are multi label classification tasks). This means that the models must be trained enough to have the insight to be able to recognize those instances that do not match with any label and thus to map them to the "No Label" pseudolabel, which in general is not an easy task.
Moreover, it's interesting to notice the highest macro F1 score achieved in the ECtHR Task A by the Legal Longformer Extra Global model (equal to 68.4) , which outperforms the Legal Longformer Extended variant by more than 4% and all other models including the standard Longformer by approximately 5%. In order to explain this great improvement of the score, we will again look at Figure 4.4 with the per label F1 scores. From this figure we observe that the main difference seems to come from label 5 where this model has an F1 score that equals 66.67 while other models score in the same label from zero to 30. Nevertheless, we have to keep in mind that label 5, that we extendedly discussed before, has an extremely small number of examples to train from and to test, which means that this extreme score improvement may be not a reliable sign of this model's capabilities in comparison with the others.

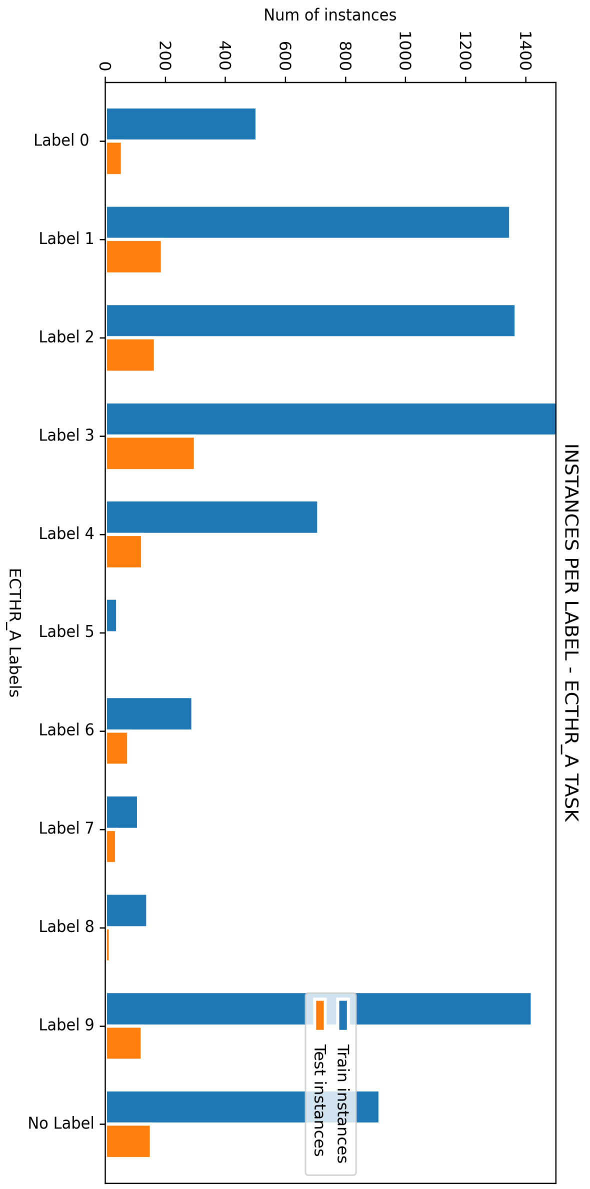
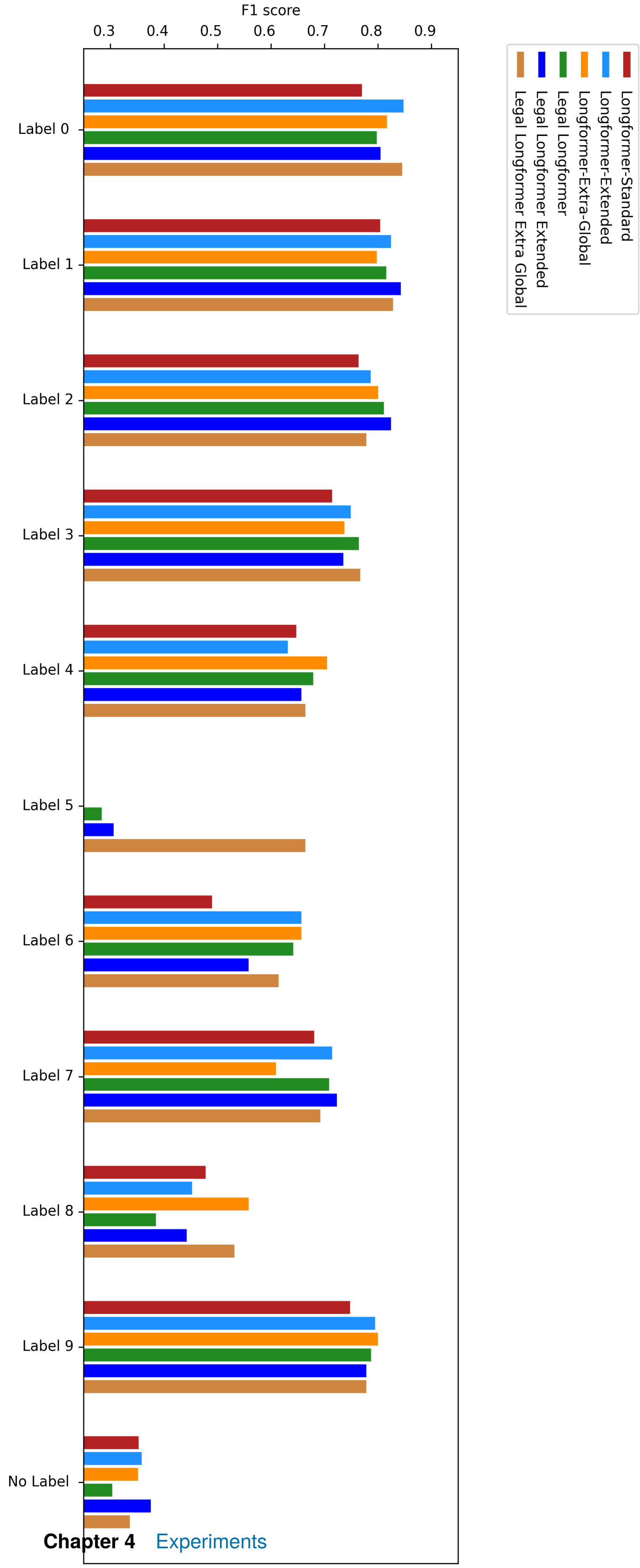


Fig. 4.3: Distribution of train and test instances in ECtHR Task A dataset across all ECHR labels. The last label called "No Label" refers to the pseudolabel that instances with no label match are mapped to. Recall that ECtHR Task A is a multi-label classification problem.

Fig. 4.4: Figure showing the F1 scores achieved by each of our implemented models in ECtHR Task A across all ECHR labels.



4.3.3 ECtHR Task B, Results & Discussion

ECtHR Task B is the next task that we evaluated our implemented methods on. This task uses exactly the same dataset with ECtHR task A, but different from the latter, this task's purpose is to map each case of the dataset to the articles of ECHR that were "allegedly" violated while task's A goal is to map each case to the articles that were truly violated. As a result, the same case examples, meaning the same texts, can be mapped to different labels in task A and task B. What is obviously expected is for task B to be much easier for the models to cope with, than task A, because predicting the allegedly violated articles from a list of factual paragraphs from each case is intuitively easier than predicting those truly violated. Table 4.5 presents the results of running all our implemented methods on ECtHR Task B. With a first look, we can see that actually all scores achieved by our models in this task are much better (around 7-10% higher average scores) comparing to task's A scores.

Starting the discussion around ECtHR task B results, we notice an increase in the scores of our non-legally pretrained methods (Longformer Extended and Longformer Extra Global) compared to the Longformer standard model. Longformer Extended achieves an increase of approximately 3% in the macro F1 score while Longformer Extra Global an increase of 2% in the same metric. From these results we conclude that these 2 models seem to take advantage of the doubled positional embeddings in ECtHR Task B, unlike the respective results of the task A and despite having an average word length that does not overcome the 4096-limit of Longformer. We also observe that the Extra Global setting is not such helpful for this task in comparison with the simple Extended setting. On the other hand, for the legally warm started models, we notice improvements of the scores achieved as we noticed in the previous two tasks. Again, this was expected, because of the fact that these 3 tasks of LexGLUE are legal tasks and legally warm started models are expected to achieve better scores than our first variants of Longformer. The best micro F1 score is achieved by the Legal Longformer Extended model, while the highest macro F1 score is achieved among all examined models by the Legal Longformer variant that has no doubled positional embeddings.

For a deeper analysis of the results, in Figure 4.6 we present the per label F1 scores in ECtHR Task B for all the examined models together with the standard Longformer. Also, following the same practice as in task A, task B uses a pseudolabel for those cases that are not matched with any ECHR article. From that figure, we notice that the general results across all labels are good, with the exception of Label 8 and the specially added "No Label". Figure 4.5 which depicts the distribution of the train/test cases among the ECHR labels in task B, shows a small distribution of cases across the labels 5,6,7,8 and "No Label". However, from the results shown in Figure 4.6 on labels 5,6,7, all models are achieving great results despite the low number of cases they have "seen" during training with the exception of

Label 8 and “No label”. That means that in comparison to task’s A performance, task B is more able to generalize across those labels with a small number of train or test instances. As for the “No Label” scores, as discussed in the previous subsection, we consider it a special added label which semantically is difficult for all models to “cope” with.

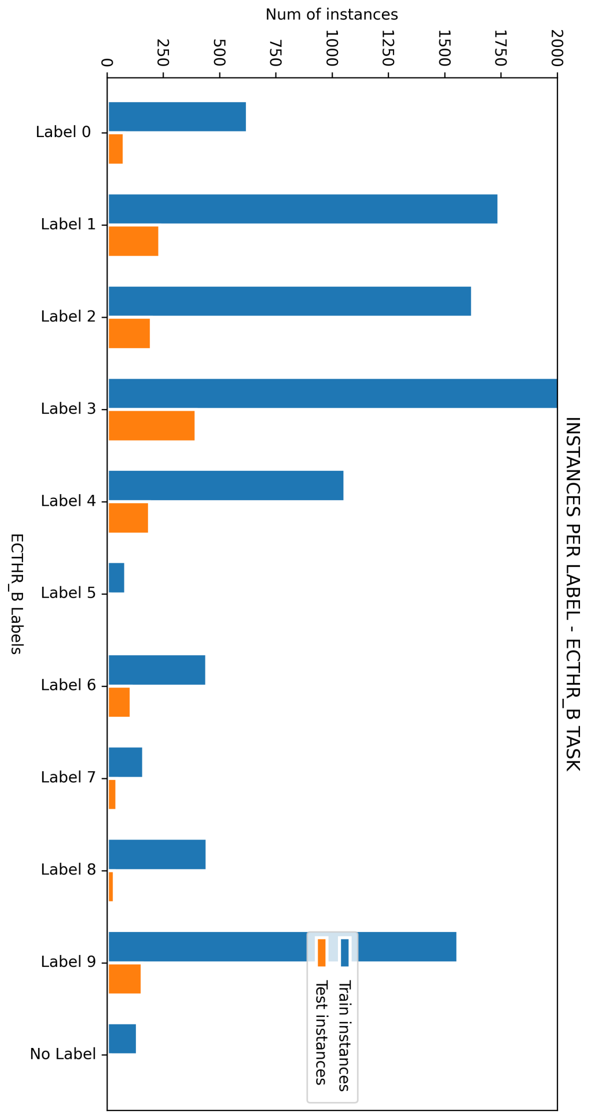
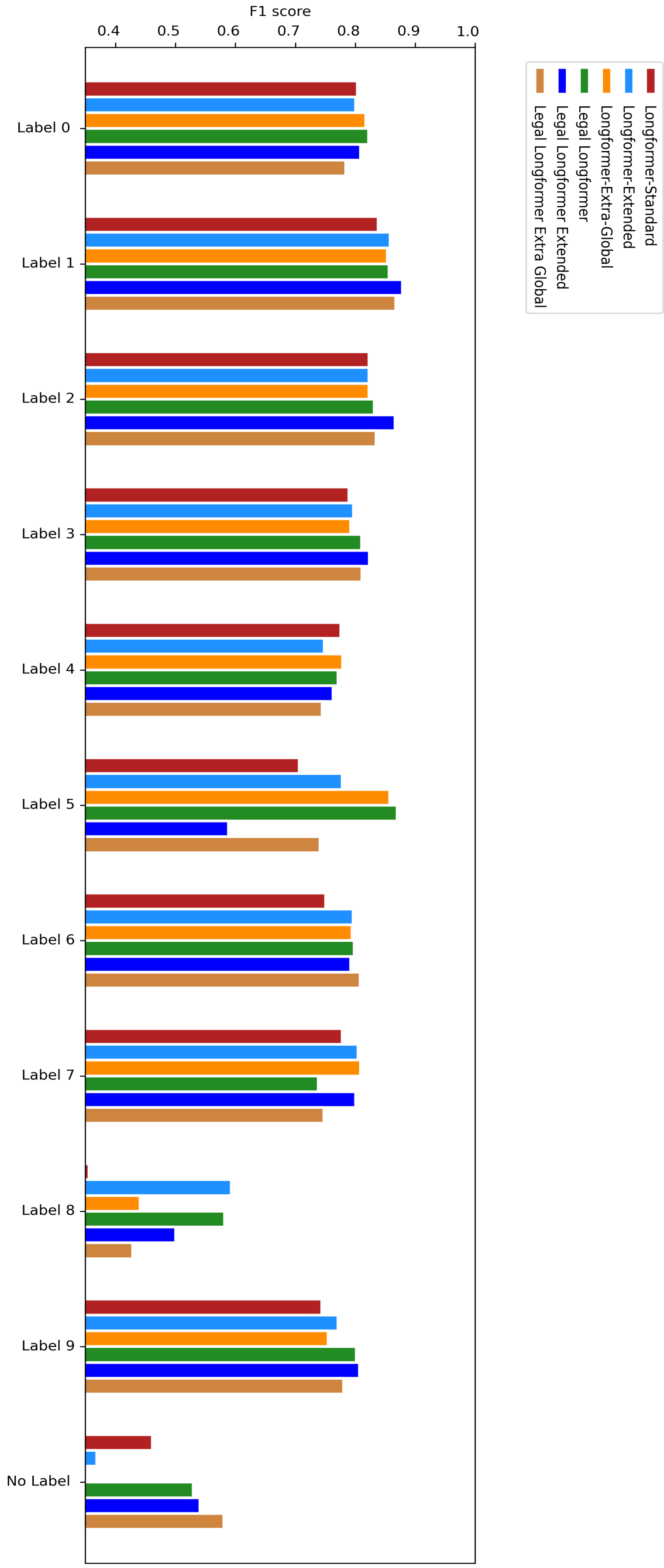


Fig. 4.5: Distribution of train and test instances in ECtHR Task(B) dataset across all ECHR labels. The last label called "No Label" refers to the pseudolabel that instances with no label match are mapped to. Recall that ECtHR task B is a multi-label classification problem.

Fig. 4.6: Figure showing the F1 scores achieved by each of our implemented models in ECtHR Task B across all ECHR labels.



5 Conclusions and Future Work

5.1 Conclusions

This thesis addressed the task of long document processing with Transformer models, which has many important applications including QA systems, text summarization, long document classifications etc. Especially, our focus was on legal NLU dataset tasks which in general consist of large texts organized in passages, paragraphs or other ways usual in the legal domain. The models we implemented were based on Longformer [BPC20], a model developed to achieve an effective processing of long and structured document tasks. Our approaches focused on modifying Longformer's components such as attention pattern, input limit length and pretraining initialization in such way that could hopefully lead to an even more effective and successful processing of long (legal) document tasks. As our focus was mostly on legal domain tasks, we experimented with the LexGLUE benchmark [Cha+22], which consists of seven legal NLU tasks, and in particular on three of its tasks consisting of large texts: SCOTUS and ECtHR Tasks A & B. Our proposed models achieve, in general, higher scores than the standard Longformer model across all 3 tasks we evaluated our methods on. In particular, we observed that "Longformer Extended" and "Longformer Extra Global" models achieve higher scores on 2 of the 3 LexGLUE tasks we experimented with, in comparison with the standard Longformer's performance, while at the same time all our models that were warm started from Legal BERT [Cha+20] show a remarkable increase in their scores comparing them with the standard Longformer and our models not warm started from Legal BERT.

5.2 Future Work

The work of this thesis leaves much space for future research and contribution. First of all, regarding our implemented methods, we aim to focus more on the models that were warm started from Legal BERT, and which showed great results across all examined tasks. In particular, continuing from these models we can apply some extra pretraining updates on legal corpora, just like Longformer does when warm started from a RoBERTa pretrained checkpoint (on general corpus data). In that way, we hope that these models can better adapt to the new architectural and attention pattern changes we have applied and achieve even better results on legal domain tasks. Moreover, another idea for future work is the application of data augmentation methods to the legal dataset tasks we focused our endeavors. These methods could help in strengthening the equal representation of all labels across the datasets of LexGLUE and as a result avoid cases of extremely low representation of labels on train and test datasets that lead to low or unstable results as we observed in some of our experimental results.

Bibliography

[Ain+20] Joshua Ainslie, Santiago Ontañón, Chris Alberti, et al. "ETC: Encoding Long and Structured Data in Transformers". In: *CoRR* abs/2004.08483 (2020). arXiv: 2004.08483.

[BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer". In: *CoRR* abs/2004.05150 (2020).

[Cha+20] 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.

[Cha+22] 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*. Dubln, Ireland, 2022.

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[Din+20] Siyu Ding, Junyuan Shang, Shuohuan Wang, et al. "ERNIE-DOC: The Retrospective Long-Document Modeling Transformer". In: *CoRR* abs/2012.15688 (2020). arXiv: 2012.15688.

[Liu+19] Yinhan Liu, Myle Ott, Naman Goyal, et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762.

[Yan+20] Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. "Beyond 512 Tokens: Siamese Multi-depth Transformer-based Hierarchical Encoder for Long-Form Document Matching". In: *CIKM*. 2020.

[Zah+20] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, et al. "Big bird: Transformers for longer sequences". In: *Advances in Neural Information Processing Systems* 33 (2020).

List of Acronyms

AI Artificial Intelligence

SoTA State of the art

NLP Natural Language Processing

NLU Natural Language Understanding

NER Name Entity Recognition

POS Part of Speech Tagging

SMITH Siamese Multi-depth Transformer-based Hierarchical

BERT Bidirectional Encoder Representations from Transformers

LexGLUE Legal General Language Understanding Evaluation

SCOTUS Supreme Court of the United States

ECHR European Court of Human Rights

ETC Extended Transformer Construction

QA Question Answering

CLS Classification

μ-F1 micro F1 score

m-F1 macro F1 score

TP True Positive

FP False Positive

FN False Negative

List of Figures

2.1 Illustration of the full n^2 attention pattern in comparison with the sliding window attention, the dilated window attention (a variant of the sliding window) and the combination of the global attention with the sliding window as implemented in Longformer. Figure taken from [BPC20]. 4

2.2 Illustration of the global-local attention mechanism that ETC implements. Figure taken from [Ain+20]. 5

2.3 An illustration of the three main components of BigBird's attention pattern. (a) shows the Random attention part, (b) shows the Window attention component, (c) shows the global attention part and (d) depicts the final combination of these 3 components that result to the final BigBird attention pattern. Figure taken from [Zah+20]. 7

2.4 Visualization of the 2-level SMITH hierarchical encoder architecture. Figure taken from [Yan+20] 8

2.5 Visualization of the general architectural organization of Ernie-DOC model including the Enhanced recurrent mechanism and the Retrospective Feed phase. Figure taken from [Din+20] 9

4.1 Distribution of train and test instances in SCOTUS dataset across all SCOTUS labels. Note that labels 12 and 13 are not shown due to their minimal or zero appearance in the dataset. 25

4.2 Figure showing the F1 scores achieved by each of our implemented models in SCOTUS task across all SCOTUS labels. 26

4.3 Distribution of train and test instances in ECtHR Task A dataset across all ECHR labels. The last label called "No Label" refers to the pseudolabel that instances with no label match are mapped to. Recall that ECtHR Task A is a multi-label classification problem. 29

4.4 Figure showing the F1 scores achieved by each of our implemented models in ECtHR Task A across all ECHR labels. 30

4.5 Distribution of train and test instances in ECtHR Task(B) dataset across all ECHR labels. The last label called "No Label" refers to the pseudolabel that instances with no label match are mapped to. Recall that ECtHR task B is a multi-label classification problem. 32
4.6 Figure showing the F1 scores achieved by each of our implemented models in ECtHR Task B across all ECHR labels. 33

List of Tables

3.1 A summary of the implemented methods presented in this section, along with their main features. Longformer-Standard refers to the original Longformer model presented in the corresponding paper [BPC20]. 15

4.1 An example of a SCOTUS text case instance together with its ground truth SCOTUS label. The text example is truncated for space purposes. 18
4.2 An ECtHR Task A case example (list of factual paragraphs) along with the ECHR Article violated. The case text is truncated for space purposes. 19
4.3 An ECtHR Task B case example instance (list of factual paragraphs) along with the ground-truth ECHR Articles that were allegedly violated. The case text is truncated for space purposes. 20
4.4 A summary of the LexGLUE task datasets including their main features and statistics. (Task Instances and Average Word Length columns refer to train/development/test subsets of each task respectively). 21
4.5 Test results for all implemented models across LexGLUE tasks, with the seed with the best development score. * In these 2 experiments, there were 2 seeds with the same macro F1 score in the development set and we reported the test results of the seed with higher test scores. 23