Long Text Classification using Transformers with Paragraph Selection Strategies

Mohit Tuteja
Thomson Reuters Labs
Bangalore, Karnataka, India
mohit.tuteja@thomsonreuters.com

Daniel González Juclà
Thomson Reuters Labs
Zug, Switzerland
daniel.gonzalezjucla@thomsonreuters.com

Abstract

In the legal domain, we often perform classification tasks on very long documents, for example court judgements. These documents often contain thousands of words, so the length of these documents poses a challenge for this modelling task. In this research paper, we present a comprehensive evaluation of various strategies to perform long text classification using Transformers in conjunction with strategies to select document chunks using traditional NLP models. We conduct our experiments on 6 benchmark datasets comprising lengthy documents, 4 of which are publicly available. Each dataset has a median word count exceeding 1,000. Our evaluation encompasses state-of-the-art Transformer models, such as RoBERTa, Longformer, HAT, MEGA and LegalBERT and compares them with a traditional baseline TF-IDF + Neural Network (NN) model. We investigate the effectiveness of pre-training on large corpora, fine-tuning strategies, and transfer learning techniques in the context of long text classification.

1 Introduction

The performance of text classification methods has improved significantly over the last decade for text instances containing less than 512 characters. Due to the high computational cost of processing longer text instances, this limitation is introduced in the most recent transformer models. To alleviate this problem and improve the classification of long texts, researchers tried to address the root causes of the computational cost and proposed the optimization of the attention mechanism (Vaswani et al., 2017), which is a key element of any transformer model.

This research undertakes a comparative study of different ways of doing long text, multi-label classification on multiple legal datasets. We take existing approaches from literature and try an alternate approach of our own. The end goal is to strike a balance between model performance and computational efficiency.

2 Related Work

Natural Language Processing (NLP) is a subfield of computer science and linguistics that focuses on processing language, including text. Probably the most basic task within this field is Text Classification, which given text as an input the objective is to classify it among a set of categories. To perform this classification, the model needs to extract features from the text and preferably to understand correlations between its different parts, in a similar way that us humans do where we connect different entities in a sentence to extract meaning from them.

Some approaches treated text with a Bag-of-Words (BoW) model, as introduced by Harris (Harris, 1954), where grammar and word order are disregarded and we consider text as a set of words with their multiplicity. Related to this concept, a popular idea to model text which became popular during the era of Statistical NLP (1990s–2010s) is TF-IDF (Term Frequency - Inverse Document Frequency). First described in 1972, TF-IDF (Sparck Jones, 1972) as its name indicates consists on vectorizing text by reflecting how important a term is to a document in a collection or corpus. Its main intuition is to divide the occurrences of the term in the text by the occurrences of this term in the whole corpus, this way we get the relative importance of a given term.

In the decade of the 2010s we dive into the era of the Neural Models for NLP, starting in 2013 with the word2vec embeddings (Mikolov et al., 2013) used to train a Recurrent Neural Network (RNN). This idea of training representation of words and terms quickly became very popular and since then there have appeared many methods to obtain these representations in an unsupervised way to then use them for downstream tasks, for example GloVe

Proceedings of the Natural Legal Language Processing Workshop 2023, pages 17–24

(Pennington et al., 2014) and fastText (Bojanowski et al., 2017).

A big revolution in NLP started with the presentation of the Transformer architecture by Vaswani (Vaswani et al., 2017) and its attention mechanism. This architecture later evolved into models that have also been pre-trained to obtain representations of the input tokens, as for example BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019). Legal-BERT (Chalkidis et al., 2020) consists in BERT heavily pre-trained with legal data, and this specialization has been proved to improve the performance of the model in downstream tasks that use this language. Nonetheless, in this study we do not include models pre-trained with legal corpora because it would include a bias when comparing with other architectures that do not have a public version pre-trained on this type of language.

The limitation of these mentioned models is that due to the quadratic computational cost the attention mechanism, they allowed up to 512 tokens in the input, while legal documents can easily be formed by thousands of them. This problem has led to two alternatives: truncation methods in order to process a part of the text, for example the first 512 tokens, and to derive the classification from that part and models that contain a more efficient attention mechanism rather than computing the pair-wise attention between all the tokens in the input.

Many sparse-attention models have been proposed, the most popular ones being Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020) and most recently Mega (Ma et al., 2022) has proved to be the best in the benchmark for long sequence modeling "Long Range Arena" (Tay et al., 2020). This latter one, despite being the best performing model for long sequences has not become very popular due to not being a public model that has been largely pre-trained. Another popular approach to process long text are Hierarchical Attention Transformers (HAT) (Pappagari et al., 2019), which aim at processing the whole text by encoding chunks of typically 512 or less tokens at each step and then encoding these together in a hierarchical way, through cross-segment attention blocks (Chalkidis et al., 2022).

Some research has been done in the direction of putting all these different methods with different intuitions in a fair comparison to solve a multi-label classification task. Some of these studies include TextGuide (Fiok et al., 2021) which compared the performance of different truncation methods in selecting the text to train a Language Model, Park (Park et al., 2022) studied and compared different efficient LMs and Chalkidis (Chalkidis et al., 2022) compared Longformer to HATs for long document classification. Nonetheless, no work has obtained definitive and actionable conclusions and more importantly no other work has compared traditional approaches such as TF-IDF vectorization followed by a Neural Network (NN) for classification with truncation methods to train popular Language Models such as BERT and Sparse-attention models designed to process long documents at once.

3 Research questions

Few works have focused on widely comparing different text classification approaches, so this method aims at setting the baseline for choosing an approach for multi-label / multi-class text classification given a set of constraints. So, the research questions that will be addressed are:

- Which is the Truncation method that works the best on input text?
- Which is the approach that reports the highest performance?
- Which is the most cost-effective approach?
- Can we train a cost-effective language model on very long sequences?

4 Datasets

Six datasets with long documents have been chosen to compare the performance of the different approaches:

Canadian Abridgement Dataset: This dataset contains 369,943 Canadian court judgements and their classification as per the Abridgement classification taxonomy as defined by the editorial team for Checkpoint. The classification follows a hierarchy, starting with 55 broad classes (subject titles). These classes have further sub-categories going up to 6 levels. Each unique classification up to the 6th level is treated as one class. The data belongs to the period 2000-2021. We have used only those cases whose classification belongs to the latest available version of the taxonomy.

Checkpoint Tax Type Classification: Thomson Reuters monitors multiple sources (tax courts, rulings, official materials etc.) in order to incorporate the latest changes in law or guidance into Checkpoint editorial content. Documents from

Dataset	Words/Doc	# Classes
Canada A	5,321	11,648
Posture 50K	2,901	256
Tax Type	708	44
SCOTUS	5,352	14
MIMIC-III	2,260	19
ECtHR	2,140	10

Table 1: Average words per document and number of unique classes for each dataset used

these sources are called Alerts. During post processing, the editors assign tags to these alerts which are known as tax-types which states what type of tax a given alert talks about. This is a multi-label classification dataset. We currently have two types of alerts - rulings and official materials.

Posture 50K: (Song et al., 2022) This data-set is publicly accessible and has been made available by Thomson Reuters. It is designed for the purpose of identifying the legal Procedural Postures involved in legal motions within a legal case. To illustrate, a plaintiff might request an appellate court to overturn a specific decision made by a lower court judge regarding a motion, which is referred to as an "On Appeal" procedural posture. The dataset comprises 50,000 legal opinions, representing real-world legal cases in the United States, along with their corresponding postures. The majority of these cases span from the year 2013 to 2020, with just three cases occurring prior to 2013.

SCOTUS: (Chalkidis et al., 2021b) This constitutes one of the six datasets within LexGLUE. It encompasses court opinions issued by the United States Supreme Court (SCOTUS). The primary challenge within this dataset involves single-label multi-class classification, with the aim of forecasting the pertinent issue area for each court opinion. There are 14 distinct classes that align with specific issue areas, collectively encompassing 278 issues centered around the subject matter of the legal dispute.

MIMIC-III Dataset: (Johnson et al., 2016) Contains approx. 50k discharge summaries from US hospitals. Each summary is annotated with one or more codes (labels) from the ICD-9 taxonomy. The input of the model is a discharge summary, and the output is the set of the relevant 1st level ICD-9 (19 in total) codes.

ECtHR (Task B): (Chalkidis et al., 2021a) This is one of the six data-sets from LexGLUE. It comprises approximately 11,000 cases sourced from the public database of the European Court of Human Rights (ECtHR). For each case, the dataset includes a collection of factual paragraphs extracted from the case description. Each case is associated with specific articles of the European Convention on Human Rights (ECHR) that were purportedly violated, as determined by the court. In terms of model input, it consists of the factual paragraph list for a given case, while the model's output comprises the set of articles that are alleged to have been violated.

The first two datasets are internal from Thomson Reuters, while the other four are publicly available. Mimic-III contains medical documents and the rest contain legal text, which is our focus area. Table 1 shows a summary of the statistics of each dataset.

5 Experiments

5.1 Setup

The 10 selected methods, which are explained in the next section, are trained for the 6 datasets with a standardized methodology.

Learning Rates: We trained each transformer model on three predefined learning rates to overcome possible variability on the optimal learning rate across models and datasets. These learning rates were 2e-05, 1e-05 and 5e-06 respectively. For each model, we selected the version which provided the best results on the respective development dataset. We then used this optimal configuration to score our test dataset. We used a standardized script for training of all models, starting with the same base model (RoBERTa) wherever possible.

Evaluation metric: The authors have chosen micro-f1 score as the metric to establish a comparison between the different approaches. This is motivated by it being the most used throughout the literature when comparing models for the task of multi-label classification.

Repeatability: We ensured that this training is also reproducible by using fixed seeds and used the same batch size for the models across datasets and the same batch-size * accumulation-steps of 8 across models. Out of the four publicly available datasets, we dropped a few observations only from ECtHR. These were observations with missing labels. Aside from that, the train-dev-test datasets were left unchanged and used as-is in each case.

Base Model selection: We use RoBERTa as the baseline transformer model instead of LegalBERT.

This was done due to the following reasons:

- LegalBERT uses LexGLUE (Chalkidis et al., 2021b) datasets during its pretraining process. Hence we observed abnormal performance gains on the ECtHR data-set which was not visible across other datasets.
- Longformer starts its training from a RoBERTa checkpoint so the comparison with RoBERTa becomes fair
- We decided to include MIMIC-III dataset for which LegalBERT might not be ideal.

It is to be noted though that we did observe 1-2% gains in F1 scores on legal datasets (other than LexGLUE) while making a switch from RoBERTa to LegalBERT.

5.2 Models

The following models were evaluated:

- **TF-IDF + Neural Net (NN):** Vectorization of each document through TF-IDF to then train a Neural Network to classify these vectors. Pre-processing included lower-casing, removing punctuation, special characters, digits and words containing less than 3 characters. We used TfidfVectorizer from sklearn with ngram range (1,2), and max features capped at 50,000. The neural network used 1 hidden layer, relu activation, dropout = .3 and batch size of 64. We iterated over 3 learning rates (.001,.002,.01) and 3 layer sizes (128,256,512). Early stopping on val-categorical-accuracy with patience = 3 was used. The same grid-search approach was considered for all data-sets and the hyper-parameters giving best micro-averaged F1 scores on the dev dataset were finally selected.
- **RoBERTa First 512 tokens:** Truncate the first 512 tokens of the input text to fine-tune pre-trained RoBERTa. It is likely that documents tend to contain important and descriptive information at the beginning, as for example abstracts or introductions.
- **RoBERTa Last 512 tokens:** Truncate the last 512 tokens of the input text to fine-tune pre-trained RoBERTa. It is possible that important information is found at the end of a document, as for example summaries or conclusions.
- **RoBERTa First & Last 256 tokens:** Concatenate the first and the last 256 tokens to obtain a chunk of 512 tokens to fine-tune pre-trained RoBERTa. Information at the beginning or at the end of a document e.g. introductions or conclusions might give more general information about the document than text in the middle, specially in a very long document.
- **RoBERTa First & Last 512 tokens:** Fine-tune a pre-trained RoBERTa model with chunks corresponding to first and last 512 of each document, by splitting them into two samples with the same labels. At inference time, aggregate the predictions for these two chunks per document, which can be done in different ways. For the results that we share in this paper, for the multi-label datasets, we take the mean probability for each class between the two chunks, and then find the best classification threshold. For the multi-class case, we replace the mean by max and select the class with the highest probability.
- **RoBERTa w/ Best paragraph selection:** This is a custom approach to obtain a **smaller and more targeted training dataset**. First, we train a traditional NN model on the tf-idf vectorization of the full training documents. Then for each training document, we do the following:
  1. Split the document into a maximum of 10 chunks of 512 tokens and predict for each of them with the previously trained model. Note: more than 10 chunks can be used, if desired.
  2. Take a dot product between the prediction (probability) vector for each chunk with the label vector for the document. Note: The label vector is already a binary vector indicating the presence of a class label.
  3. Select the chunk having the highest similarity with the label vector, which is likely the "most important chunk", as it has been classified the best.

  Having a dataset of the most important chunk per document, we now fine-tune a RoBERTa model for the multi-label classification task. At **inference time**, it is not possible to select the most important chunk for each document because we have no labels, so for each document, we:
  1. Split the text into a maximum of 10 chunks and predict for each of them inde-

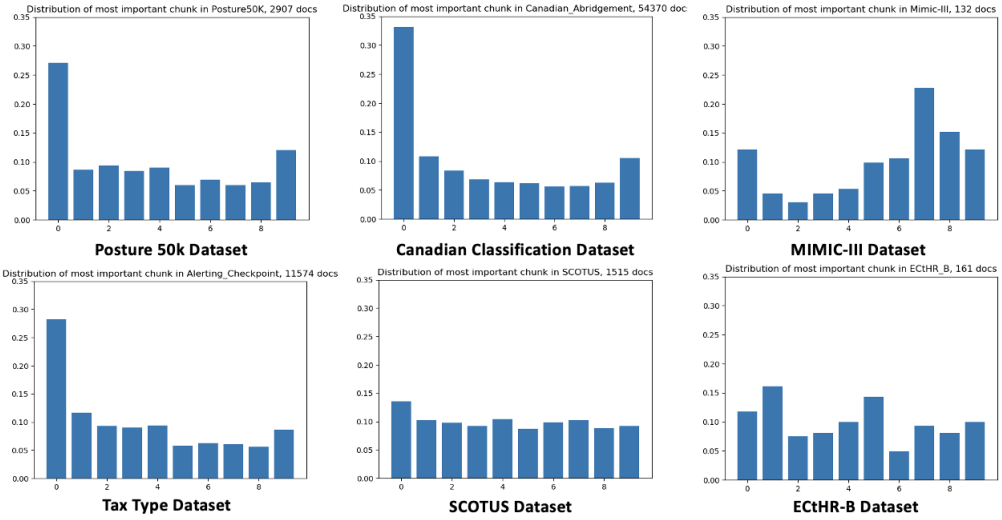

Figure 1: Paragraph importance distribution for all dev datasets. We take only those documents where we have >=10 chunks for this representation

pendently with the fine-tuned RoBERTa model.

2. The predictions for each of these chunks are then combined. The combination can be done in multiple ways. One approach could be taking a sum/average of the predictions (probabilities) for each class across chunks and then applying some threshold. Another could be taking the highest probability for each class observed across chunks and then applying a threshold. There were multiple approaches like these that were evaluated.
3. It has been found empirically that the most effective way to combine the predictions is to first predict the class that has the highest weighted average confidence. The weights in this case come from the paragraph importance distribution given in Figure 1 and derived from the model training data set. Subsequently, we predict all the other classes whose average confidence is within a ratio of 0.8 to the top class. For the multi-class classification case, we stop at only 1 prediction.

- **Longformer First 4096 tokens:** Fine-tune a pre-trained Longformer with the first 4096 tokens of the document.
- **Longformer First & Last 512: tokens:** Concatenate the first and the last 512 tokens to obtain a chunk of 1024 tokens to fine-tune a pre-trained Longformer. Same intuition as with RoBERTa and possible efficiency boost compared to Longformer processing 4096 tokens.
- **HATs First 4096 tokens:** Fine-tune a pre-trained HAT with the first 4096 tokens of the document. The chosen implementation uses BERT as a backbone model.
- **HATs First & Last 512 tokens:** Concatenate the first and the last 512 tokens to obtain a chunk of 1024 tokens to fine-tune a pre-trained HAT.
- **MEGA: Moving Average Equipped Gated Attention 4096:** Fine-tune a pre-trained (by us) MEGA with the first 4096 tokens of the document. Pre-training details can be found in appendix A.

6 Results

6.1 Paragraph importance

In the explanation of the "RoBERTa w/ Best para selection" approach it has been mentioned that during the training phase of the model we select the chunk within each document that has been predicted the best, which we call the most important. If we select the documents that were split into 10 chunks (the maximum allowed, 5 chunks at the beginning and 5 at the end) and we plot the histogram of which was the **most important paragraph**, we get the results in Figure 1.

Figure 1 shows how the first paragraph is the most important in the three Thomson Reuters datasets. In MIMIC-III though, the chunks at the end have more importance than the ones at the

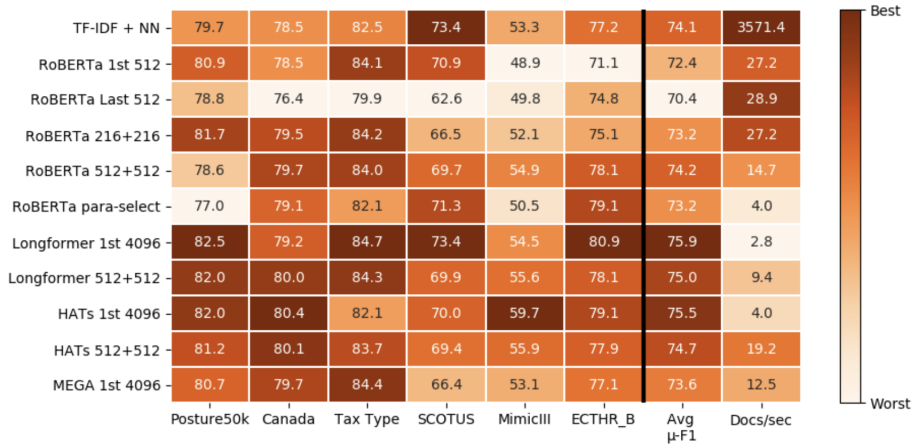

Figure 2: Heat-map of testing Micro-F1s per dataset. Inference time for TF-IDF+DNN corresponds to CPU evaluation, while the rest have been evaluated on a single GPU.

beginning. In the LexGLUE datasets, it is more evenly distributed. In the next section, it is seen how these **distributions relate to the performance of the approaches** that truncate different parts of the text.

6.2 Comparison of testing results

Figure 2 contains the testing Micro-F1 score at inference time for each combination of dataset and approach, together with a color coding to making it easier to spot the best and worst performing approaches for each dataset. These are the main observations:

- There is **not one clear model that performs the best** across all datasets, even though some approaches perform consistently better than others. In 4 out of the 6 datasets, Longformer trained with the first 4096 tokens provides the best performance, but on the other hand it is the least efficient.

- **TF-IDF + DNN provides a strong baseline**, as it is not far from the rest of RoBeRTa-based approaches and it is hundreds of times more efficient (using a CPU, while RoBERTa-based are evaluated using a single GPU).

- Models that truncate first tokens work well on Thomson Reuters datasets because of the importance of the first chunk, as seen in the "Paragraph Importance" section,

- Mimic-III and ECtHR-B are the only two datasets where the first chunk is not the most important, as it has been seen in the previous section, hence RoBERTa with first 512 tokens shows worse performance.

- **RoBERTa concatenating first and last 256 tokens** provides a better performance than RoBERTa first 512 on 5 out of 6 datasets, so in cases where it is possible to have relevant information at the end of the document, it **can be a better approach**.

- The versions of **Longformer and HAT** concatenating **first and last 512 tokens** provide a **comparable performance** to their version with the first 4096 tokens, while being around 4 times more efficient.

- Longformer performs better than HAT for 4 out of 6 datasets, but it is less efficient. Their versions concatenating the first and last 512 tokens also perform very similar.

- The in-house pretrained **MEGA has a consistently good performance** for all datasets without being the best in any, but it is considerably more efficient than the other two models with 4096 tokens.

7 Conclusions and Future Work

We recommend the following Model training Guidelines:

1. Train a baseline model (TF-IDF + DNN). It is the most efficient and typically reports good performance. This could be a good solution if 1-2% lower performance compared to SOTA is acceptable.

2. Plot the paragraph importance for your dataset, and identify which paragraphs are the most important. In case the Paragraph Importance has a U shape, like court judgement datasets, the best choice is usually a concatenated model.
    - LegalBERT/RoBERTa concatenate first and last 256, if good efficiency is needed.
    - Longformer/HAT concatenate first and last 512, if efficiency is not the priority.
3. In case Paragraph importance is more uniform, best choice is a Long model, as all paragraphs are important:
    - Longformer with first 4096 tokens will probably report the best performance, but it is expensive to train.
    - HAT Concat first and last 512 will probably report a slightly lower performance than Longformer 4096, while being quite more efficient.
    - LegalBERT concat or LegalBERT truncate will probably still report decent results, while being efficient.

Future work could be done in this topic by comparing the presented approaches to newer models that have come out after this study has been done. Some of these models can be:

- Llama 2 (Touvron et al., 2023) is a family of LLMs released by Meta with a context of 4096 tokens. Their extensive pre-training, their open-source availability and their large context length makes them a good option to be added to the comparison.

- LongNet (Ding et al., 2023) is an architecture published by Microsoft Research, whose authors claim to be able to process up to a million tokens. This is a very interesting feature since it would be able to entirely process any document in our datasets.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022. An exploration of hierarchical attention transformers for efficient long document classification. *arXiv preprint arXiv:2210.05529*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021a. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 226–241, Online. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2021b. Lexglue: A benchmark dataset for legal language understanding in english. *arXiv preprint arXiv:2110.00976*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. 2023. Longnet: Scaling transformers to 1,000,000,000 tokens.

Krzysztof Fiok, Waldemar Karwowski, Edgar Gutierrez-Franco, Mohammad Reza Davahli, Maciej Wilamowski, Tareq Ahram, Awad Al-Juaid, and Jozef Zurada. 2021. Text guide: improving the quality of long text classification by a text selection method based on feature importance. *IEEE Access*, 9:105439–105450.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. 2022. Mega: moving average equipped gated attention. *arXiv preprint arXiv:2209.10655*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 838–844. IEEE.

Hyunji Hayley Park, Yogarshi Vyas, and Kashif Shah. 2022. Efficient classification of long documents using transformers. *arXiv preprint arXiv:2203.11258*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Dezhao Song, Andrew Vold, Kanika Madan, and Frank Schilder. 2022. Multi-label legal document classification: A deep learning-based approach with label-attention and domain-specific pre-training. *Information Systems*, 106:101718.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Hyperparameters	
Datasets	Bookcorpus + CC News + Wikipedia (20220301.en)
Max sequence length	256
Training batch size	32
Accumulation steps	16
Training steps	77k
Learning rate	0.001
Weight decay	0.05
Learning rate warmup %	0.006
Encoder depth	6
Encoder embedding dim	256
Encoder Z dim	128
Encoder hidden dim	512
Dropout	0.1
Activation function	Silu

Table 2: MEGA pretraining hyperparameters

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

A MEGA pretraining

Moving Average Gated Attention (MEGA) is an architecture that has demonstrated state-of-the-art performance in long sequence modelling, given that it has the best results for the Long Range Arena at the time the research has been done.

Also at this time there was not any available pretrained version of MEGA as it was released very recently. Nonetheless, there was a public implementation, so we decided to pretrain this model on the Masked Language Modeling task in order to later fine-tune it for the multi-label classification task in the 6 different datasets.

The parameters used for MEGA pretraining can be found down in table 2.