Revisiting Transformer-based Models for Long Document Classification

Anonymous ACL submission

Abstract

The recent literature in text classification is biased towards short text sequences (e.g., sentences or paragraphs). In real-world applications, multi-page multi-paragraph documents are common and they cannot be efficiently encoded by vanilla Transformer-based models. We compare different Transformer-based Long Document Classification (TrLDC) approaches that aim to mitigate the computational overhead of vanilla transformers to encode much longer text, namely sparse attention and hierarchical encoding methods. We examine several aspects of sparse attention (e.g., size of local attention window, use of global attention) and hierarchical (e.g., document splitting strategy) transformers on four document classification datasets covering different domains. We observe a clear benefit from being able to process longer text, and, based on our results, we derive practical advice of applying Transformer-based models on long document classification tasks.

1 Introduction

Natural language processing has been revolutionised by the large scale self-supervised pre-training of language encoders (Devlin et al., 2019; Liu et al., 2019), which are fine-tuned in order to solve a wide variety of downstream classification tasks. However, the recent literature in text classification mostly focuses on short sequences, such as sentences or paragraphs (Sun et al., 2019; Adhikari et al., 2019; Mosbach et al., 2021), which are sometimes misleadingly named as documents.[1]

The transition from short-to-long document classification is non-trivial. One challenge is that BERT and most of its variants are pre-trained on sequences containing up-to 512 tokens, which is not a long document. A common practice is to truncate actually long documents to the first 512 tokens, which allows the immediate application of these pre-trained models (Adhikari et al., 2019; Chalkidis et al., 2020). We believe that this is an insufficient approach for long document classification because truncating the text may omit important information, leading to poor classification performance (Figure 1). Another challenge comes from the computational overhead of vanilla Transformer: in the multi-head self-attention operation (Vaswani et al., 2017), each token in a sequence of n tokens attends to all other tokens. This results in a function that has $O(n^2)$ time and memory complexity, which makes it challenging to efficiently process long documents.

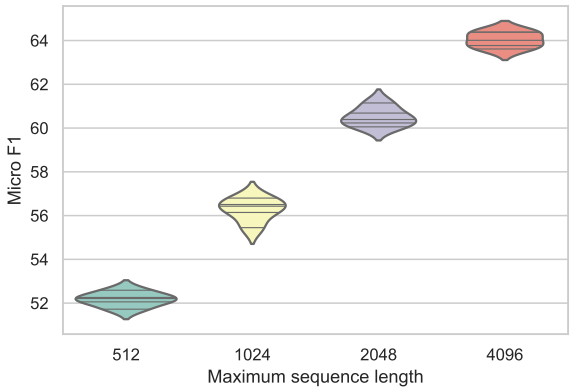


Figure 1: The effectiveness of Longformer, a long-document Transformer, on the MIMIC-III development set. There is a clear benefit from being able to process longer text.

In response to the second challenge, long-document Transformers have emerged to deal with long sequences (Beltagy et al., 2020; Zaheer et al., 2020). However, they experiment and report results on non-ideal long document classification datasets, i.e., documents on the IMDB dataset are not really long – fewer than 15% of examples are longer than 512 tokens; while the Hyperpartisan dataset only has very few (645 in total) documents. On datasets with longer documents, such as the MIMIC-III dataset (Johnson et al., 2016) with an

[1] For example, many biomedical datasets use 'documents' from the PubMed collection of biomedical literature, but these documents actually consist of titles and abstracts.

average length of 2,000 words, it has been shown that multiple variants of BERT perform worse than a CNN or RNN-based model (Chalkidis et al., 2020; Vu et al., 2020; Dong et al., 2021; Ji et al., 2021a; Gao et al., 2021; Pascual et al., 2021). We believe there is a need to understand the performance of Transformer-based models on classifying documents that are actually long.

In this work, we aim to transfer the success of the pre-train–fine-tune paradigm to long document classification. Our main contributions are:

- We compare different long document classification approaches based on transformer architecture: namely, sparse attention, and hierarchical methods. Our results show that processing more tokens can bring drastic improvements comparing to processing up-to 512 tokens.
- We conduct careful analyses to understand the impact of several design options on both the effectiveness and efficiency of different approaches. Our results show that some design choices (e.g., size of local attention window in sparse attention method) can be adjusted to improve the efficiency without sacrificing the effectiveness, whereas some choices (e.g., document splitting strategy in hierarchical method) vastly affect effectiveness.
- Last but not least, our results show that, contrary to previous claims, Transformer-based models can outperform former state-of-the-art CNN based models on MIMIC-III dataset .

2 Problem Formulation and Datasets

We divide the document classification model into two components: (1) a document encoder, which builds a vector representation of a given document; and, (2) a classifier that predicts a single or multiple labels given the encoded vector. In this work, we mainly focus on the first component: we use Transformer-based encoders to build a document representation, and then take the encoded document representation as the input to a classifier. For the second component, we use a TANH activated hidden layer, followed by the output layer. Output probabilities are obtained by applying a SIGMOID (multi-label) or SOFTMAX (multi-class) function to output logits.[2]

[2]Long document classification datasets are usually annotated using a large number of labels. Studies that have focused on the second component investigate methods of utilising label hierarchy (Chalkidis et al., 2020; Vu et al., 2020), pre-training label embeddings (Dong et al., 2021), to name but a few.

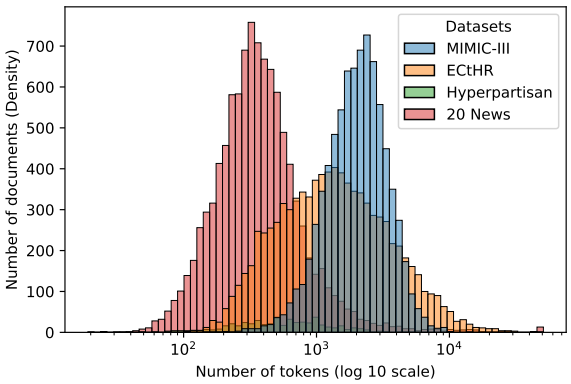


Figure 2: The distribution of document lengths. A log-10 scale is used for the X axis.

We mainly conduct our experiments on the MIMIC-III dataset (Johnson et al., 2016), where researchers still fail to transfer "the Magic of BERT" to medical code assignment tasks (Ji et al., 2021a; Pascual et al., 2021).

MIMIC-III contains Intensive Care Unit (ICU) discharge summaries, each of which is annotated with multiple labels—*diagnoses* and *procedures*—using the ICD-9 (The International Classification of Diseases, Ninth Revision) hierarchy. Following Mullenbach et al. (2018), we conduct experiments using the top 50 frequent labels.[3]

To address the generalisation concern, we also use three datasets from other domains: ECtHR (Chalkidis et al., 2022) sourced from legal cases, Hyperpartisan (Kiesel et al., 2019) and 20 News (Joachims, 1997), both from news articles.

ECtHR contains legal cases from The European Court of Human Rights' public database. The court hears allegations that a state has breached human rights provisions of the European Convention of Human Rights, and each case is mapped to one or more *articles* of the convention that were *allegedly* violated.[4]

Hyperpartisan contains news articles which are manually labelled as hyperpartisan (taking an extreme left or right standpoint) or not.[5]

[3]Details about dataset split and labels can be found at https://github.com/jamesmullenbach/caml-mimic

[4]https://huggingface.co/datasets/ecthr_cases

[5]https://pan.webis.de/semeval19/semeval19-web/; we use the split provided by Beltagy et al. (2020).

20 News contains newsgroups posts which are categorised into 20 topics.[6]

We note that documents in MIMIC-III and ECtHR are much longer than those in Hyperpartisan and 20 News (Table 4 in Appendix and Figure 2).

3 Approaches

In the era of Transformer-based models, we identify two representative approaches of processing long documents in the literature that either acts as an inexpensive drop-in replacement for the vanilla self-attention (i.e., *sparse* attention) or builds a task-specific architecture (i.e., *hierarchical* Transformers).

3.1 Sparse-Attention Transformers

Vanilla transformer relies on the multi-head self-attention mechanism, which scales poorly with the length of the input sequence, requiring quadratic computation time and memory to store all scores that are used to compute the gradients during back-propagation (Qiu et al., 2020). Several Transformer-based models (Kitaev et al., 2020; Tay et al., 2020; Choromanski et al., 2021) have been proposed exploring efficient alternatives that can be used to process long sequences.

Longformer of Beltagy et al. (2020) consists of local (window-based) attention and global attention that reduces the computational complexity of the model and thus can be deployed to process up to 4096 tokens. Local attention is computed in-between a window of neighbour (consecutive) tokens. Global attention relies on the idea of global tokens that are able to attend and be attended by any other token in the sequence (Figure 7 in Appendix). **BigBird** of Zaheer et al. (2020) is another sparse-attention based Transformer that uses a combination of a local, global and random attention, i.e., all tokens also attend a number of random tokens on top of those in the same neighbourhood. Both models are warm-started from the public RoBERTa checkpoint and are further pre-trained on masked language modelling. They have been reported to outperform RoBERTa on a range of tasks that require modelling long sequences.

We choose Longformer (Beltagy et al., 2020) in this study and refer readers to Xiong et al. (2021) for a systematic comparison of recent proposed efficient attention variants.

[6] http://qwone.com/~jason/20Newsgroups/

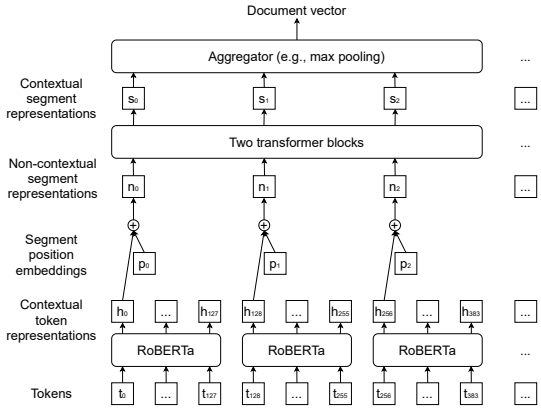


Figure 3: A high-level illustration of hierarchical Transformers. A shared pre-trained RoBERTa is used to encode each segment, and a two layer transformer blocks is used to capture the interaction between different segments. Finally, contextual segment representations are aggregated into a document representation.

3.2 Hierarchical Transformers

Instead of modifying multi-head self-attention mechanism to efficiently model long sequences, hierarchical Transformers build on top of vanilla transformer architecture.

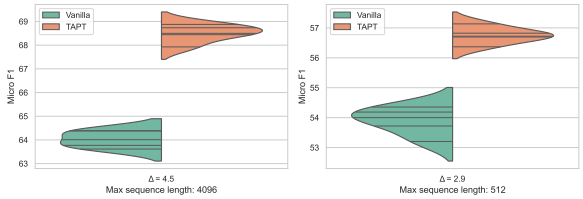
A document, $\mathcal{D} = \{t_0, t_1, \cdots, t_{|\mathcal{D}|}\}$, is first split into segments, each of which should have less than 512 tokens. These segments can be independently encoded using any pre-trained Transformer-based encoders (e.g., RoBERTa in Figure 3). We sum the contextual representation of the first token from each segment up with segment position embeddings as the segment representation (i.e., n_i in Figure 3). Then the segment encoder—two transformer blocks (Zhang et al., 2019)—are used to capture the interaction between segments and output a list of contextual segment representations (i.e., s_i in Figure 3), which are finally aggregated into a document representation. By default, the aggregator is the **max-pooling** operation unless other specified.[7]

4 Experimental Setup

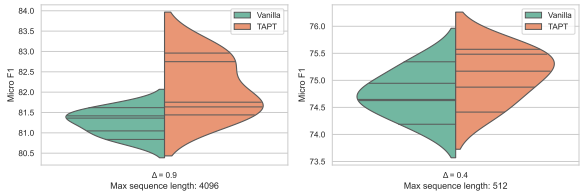
Backbone Models We mainly consider two models in our experiments: Longformer-base (Beltagy et al., 2020), and RoBERTa-base (Liu et al., 2019) which is used in hierarchical Transformers.

Evaluation metrics For the MIMIC-III (multilabel) dataset, we follow previous work (Mullenbach et al., 2018; Cao et al., 2020) and use micro-averaged AUC (Area Under the receiver

[7] Code is available at URL



(a) Longformer on MIMIC-III (b) RoBERTa on MIMIC-III



(c) Longformer on ECtHR (d) RoBERTa on ECtHR

Figure 4: Task-adaptive pre-training (right side in each plot) can improve the effectiveness (measured on the development sets) of pre-trained models by a large margin on MIMIC-III, but small on ECtHR. Δ: the difference between mean values of compared experiments.

operating characteristic Curve), macro-averaged AUC, micro-averaged F_1, macro-averaged F_1 and Precision@5—the proportion of the ground truth labels in the top-5 predicted labels—as the metrics. We report micro and macro averaged F_1 for the ECtHR (multilabel) dataset, and accuracy for both Hyperpartisan (binary) and 20 News (multiclass) datasets.

5 Experiments

We conduct a series of controlled experiments to understand the impact of design choices in different TrLDC models. Bringing these optimal choices all together, we compare TrLDC against the state of the art, as well as baselines that only process up-to 512 tokens. Finally, based on our investigation, we derive practical advice of applying transformer-based models to long document classification regarding both effectiveness and efficiency.

Task-adaptive pre-training is a promising first step. Domain-adaptive pre-training (DAPT) – the continued pre-training a language model on a large corpus of domain-specific text – is known to improve downstream task performance (Gururangan et al., 2020; Kær Jørgensen et al., 2021). However, task-adaptive pre-training (TAPT) – continues unsupervised pre-training on the task's data – is comparatively less studied, mainly because most of the benchmarking corpora are small and thus the benefit of TAPT seems less obvious than DAPT.

We believe document classification datasets, due to their relatively large size, can benefit from TAPT. On both MIMIC-III and ECtHR, we continue to pre-train Longformer and RoBERTa using the masked language modelling pre-training objective (details about pre-training can be found at Appendix 9.4). We find that task-adaptive pre-trained models substantially improve performance on MIMIC-III (Figure 4 (a) and (b)), but there are smaller improvements on ECtHR (Figure 4 (c) and (d)). We suspect this difference is because legal cases (i.e., ECtHR) are publicly available and have been covered in pre-training data used for training Longformer and RoBERTa, whereas clinical notes (i.e., MIMIC-III) are not (Dodge et al., 2021). See Appendix 9.6 for a short analysis on this matter.

We also compare our TAPT-RoBERTa against publicly available domain-specific RoBERTa, trained from scratch on biomedical articles and clinical notes. Results (Figure 8 in Appendix) show that TAPT-RoBERTa outperforms domain-specific base model, but underperforms the larger model.

Size	Micro F_1	Speed	
		Train	Test
32	67.9 $\pm$ 0.3	9.9 (2.9x)	45.6 (2.8x)
64	68.1 $\pm$ 0.1	8.8 (2.6x)	41.4 (2.5x)
128	68.3 $\pm$ 0.3	7.4 (2.1x)	34.1 (2.1x)
256	68.4 $\pm$ 0.3	5.5 (1.6x)	25.4 (1.6x)
512	68.5 $\pm$ 0.3	3.5 (1.0x)	16.3 (1.0x)

Table 1: The impact of local attention window size in Longformer on MIMIC-III development set. Speed is measured using 'processed samples per second', and numbers in parenthesis are the relative speedup.

5.1 Longformer

Small local attention windows are effective and efficient. Beltagy et al. (2020) observe that many tasks do not require reasoning over the entire context. For example, they find that the distance between any two mentions in a coreference resolution dataset (i.e., OntoNotes) is small, and it is possible to achieve competitive performance by processing small segments containing these mentions.

Inspired by this observation, we investigate the impact of local context size on document classification, regarding both effectiveness and efficiency. We hypothesise that long document classification, which is usually paired with a large label space, can be performed by models that only attend over short sequences instead of the entire document (Gao

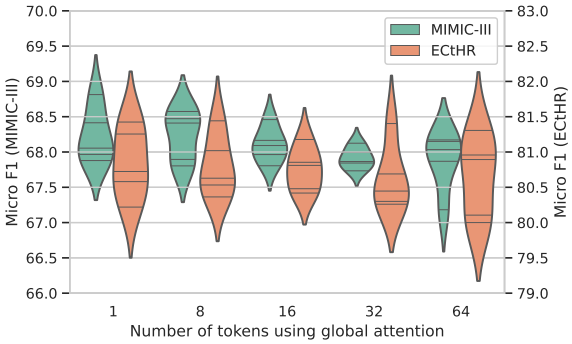


Figure 5: The effect of applying global attention on more tokens, which are evenly chosen based on their positions. In the baseline model (first column), only the [CLS] token uses global attention.

et al., 2021). In this experiment, we vary the local attention window around each token.

Table 1 shows that even using a small window size, the micro F_1 score on MIMIC-III development set is still close to using a larger window size. We observe the same pattern on ECtHR and 20 News (See Table 11 in the Appendix). A major advantage of using smaller local attention windows is the faster computation for training and evaluation.

Considering a small number of tokens for global attention improves the stability of the training process. Longformer relies heavily on the [CLS] token, which is the only token with global attention—attending to all other tokens and all other tokens attending to it. We investigate whether allowing more tokens to use global attention can improve model performance, and if yes, how to choose which tokens to use global attention.

Figure 5 shows that adding more tokens using global attention does not improve F_1 score, while a small number of additional global attention tokens can make the training more stable.

Equally distributing global tokens across the sequence is better than content-based attribution. We consider two approaches to choose additional tokens that use global attention: position based or content based. In the position-based approach, we distribute n additional tokens at equal distances. For example, if $n = 4$ and the sequence length is 4096, there are global attention on tokens at position 0, 1024, 2048 and 3072. In the content-based approach, we identify informative tokens, using TF-IDF (Term Frequency–Inverse Document Frequency) within each document, and we apply global attention on the top-K informative tokens, together with the [CLS] token. Results show that

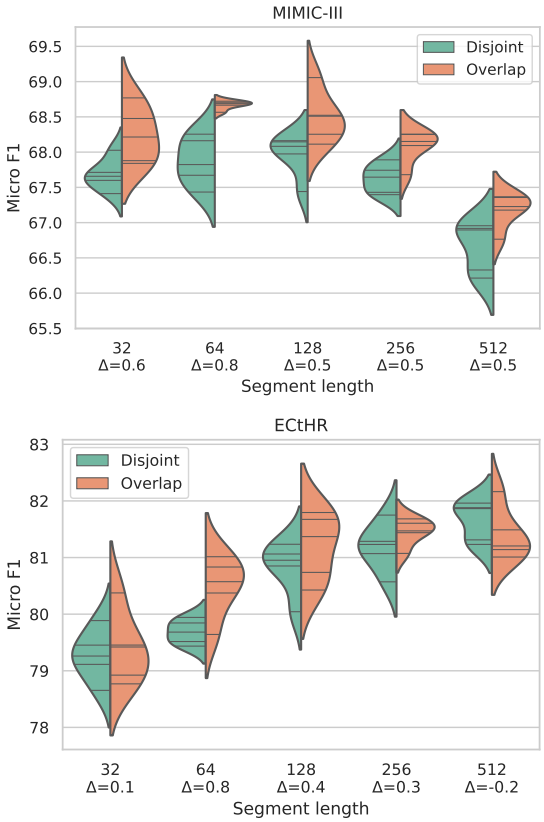


Figure 6: The effect of varying the segment length and whether allowing segments to overlap in the hierarchical Transformers. Δ: improvement due to overlap.

the position based approach is more effective than content based (see Table 13 in the Appendix).

5.2 Hierarchical Transformers

The optimal segment length is dataset dependent. Ji et al. (2021a) and Gao et al. (2021) reported negative results with a hierarchical Transformer with a segment length of 512 tokens on the MIMIC-III dataset. Their methods involved splitting a document into equally sized segments, which were processed using a shared BERT encoder. Instead of splitting the documents into such large segments, we investigate the impact of segment length and preventing context fragmentation.

Figure 6 (left side in each violin plot) shows that there is no optimal segment length across both MIMIC-III and ECtHR. Small segment length works well on MIMIC-III, and using segment length greater than 128 starts to decrease the performance. In contrast, the ECtHR dataset benefits from a model with larger segment lengths. The optimal performing segment length on 20 News and Hyperpartisan are 256 and 128, respectively (See Table 14 in the Appendix).

Splitting documents into overlapping segments can alleviate the context fragmentation problem. Splitting a long document into smaller segments may result in the problem of context fragmentation, where a model lacks the information it needs to make a prediction (Dai et al., 2019; Ding et al., 2021). Although, the hierarchical model uses a second-order transformer to fuse and contextualise information across segments, we investigate a simple way to alleviate context fragmentation by allowing segments to overlap when we split a document into segments. That it, except for the first segment, the first $\frac{1}{4}n$ tokens in each segment are taken from the previous segment, where n is the segment length. Figure 6 (right side in each violin plot) show that this simple strategy can easily improve the effectiveness of the model.

Splitting based on document structure. Chalkidis et al. (2022) argue that we should follow the structure of a document when splitting it into segments (Tang et al., 2015; Yang et al., 2016). They propose a hierarchical Transformer for the ECtHR dataset that splits a document at the paragraph level, reading up to 64 paragraphs of 128 token each (8192 tokens in total).

We investigate whether splitting based on document structure is better than splitting a long document into segments of same length. Similar to their model, we consider each paragraph as a segment and all segments are then truncated or padded to the same segment length. We follow Chalkidis et al. (2022) and use segment length (l) of 128 on ECtHR, and tune $l \in \{32, 64, 128\}$ on MIMIC-III.[8]

Results show that splitting by the paragraph-level document structure does not improve performance on the ECtHR dataset. On MIMIC-III, splitting based on document structure substantially underperforms evenly splitting the document (Figure 9 in the Appendix) .

5.3 Label-wise Attention Network

Recall from Section 3 that our models form a single document vector which is used for the final prediction. That is, in Longformer, we use the hidden states of the [CLS] token; in hierarchical models, we use the max pooling operation to aggregate a list of contextual segment representations into a document vector. The Label-Wise Attention Network (LWAN) (Mullenbach et al., 2018; Xiao et al., 2019; Chalkidis et al., 2020) is an alternative that allows the model to learn distinct document representations for each label. Given a sequence of hidden representations (e.g., contextual token representations in Longformer or contextual segment representations in hierarchical models: $\boldsymbol{S} = [\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_m]$), LWAN can allow each label to learn to attend to different positions via:

$$\boldsymbol{a}_\ell = \text{SoftMax}(\boldsymbol{S}^\top \boldsymbol{u}_\ell) \quad (1)$$

$$\boldsymbol{v}_\ell = \sum_{i=1}^{m} \boldsymbol{a}_{\ell,i} \boldsymbol{s}_i \quad (2)$$

$$\hat{\boldsymbol{y}}_\ell = \sigma(\boldsymbol{\beta}_\ell^\top \boldsymbol{v}_\ell) \quad (3)$$

where $\boldsymbol{u}_\ell$ and $\boldsymbol{\beta}_\ell$ are vector parameters for label ℓ.

Results show that adding a LWAN improves performance on MIMIC-III (Micro F_1 score of 1.1 with Longformer; 1.8 with hierarchical models), where on average each document is assigned 6 labels out of 50 available labels (classes). There is a smaller improvement on ECtHR (0.4 with Longformer; 0.1 with hierarchical models), where the average number of labels per document is 1.5 out of 10 labels (classes) in total (Table 16 in the Appendix).

5.4 Comparison with State of the art

We compare TrLDC models against recently published results on MIMIC-III, as well as baseline models that process up to 512 tokens. In addition to the common practice of truncating long documents (i.e., using the first 512 tokens), we consider two alternatives that either randomly choose 512 tokens from the document as input or take as input the most informative 512 tokens, identified using TF-IDF scores.

Results in Table 2 and 3 show that there is a clear benefit from being able to process longer text. Both the Longformer and hierarchical Transformers outperform baselines that process up to 512 tokens with a large margin on MIMIC-III and ECtHR, whereas relatively small improvements on 20 News and Hyperpartisan. It is also worthy noting that, among these baselines, there is no single best strategy to choose which 512 tokens to process. Using the first 512 tokens works well on MIMIC-III and Hyperpartisan datasets, but it performs much worse than 512 random tokens on ECtHR.

Finally, Longformer, which can process up to 4096 tokens, achieves competitive results with the

[8] Note that since we need to pad short segments, therefore, a larger maximum sequence length is required to preserve the same information as in evenly splitting.

		Macro AUC	Micro AUC	Macro F_1	Micro F_1	P@5
CAML (Mullenbach et al., 2018)	$\mathbb{C}$	88.4	91.6	57.6	63.3	61.8
PubMedBERT (Ji et al., 2021a)	$\mathbb{T}$	88.6	90.8	63.3	68.1	64.4
GatedCNN-NCI (Ji et al., 2021b)	$\mathbb{C}$	91.5	93.8	62.9	68.6	65.3
LAAT (Vu et al., 2020)	$\mathbb{R}$	92.5	94.6	66.6	71.5	67.5
MSMN (Yuan et al., 2022)	$\mathbb{R}$	**92.8**	**94.7**	**68.3**	**72.5**	**68.0**
Baselines processing up to 512 tokens						
First	$\mathbb{T}$	83.0 $\pm$ 0.1	86.0 $\pm$ 0.1	47.0 $\pm$ 0.4	56.1 $\pm$ 0.2	55.4 $\pm$ 0.2
Random	$\mathbb{T}$	82.5 $\pm$ 0.2	85.4 $\pm$ 0.1	42.7 $\pm$ 0.4	51.1 $\pm$ 0.2	52.3 $\pm$ 0.2
Informative	$\mathbb{T}$	82.7 $\pm$ 0.1	85.8 $\pm$ 0.1	46.4 $\pm$ 0.5	55.2 $\pm$ 0.3	54.8 $\pm$ 0.2
Long document models						
Longformer (4096 + LWAN)	$\mathbb{T}$	90.0 $\pm$ 0.1	92.6 $\pm$ 0.2	60.7 $\pm$ 0.6	68.2 $\pm$ 0.2	64.8 $\pm$ 0.2
Hierarchical (4096 + LWAN)	$\mathbb{T}$	91.1 $\pm$ 0.1	93.6 $\pm$ 0.0	62.9 $\pm$ 0.1	69.5 $\pm$ 0.1	65.7 $\pm$ 0.2
Hierarchical (4096 + LWAN + L*)	$\mathbb{T}$	91.7 $\pm$ 0.1	94.1 $\pm$ 0.0	65.2 $\pm$ 0.2	71.0 $\pm$ 0.1	66.2 $\pm$ 0.1
Hierarchical (8192 + LWAN)	$\mathbb{T}$	91.4 $\pm$ 0.0	93.7 $\pm$ 0.1	63.8 $\pm$ 0.3	70.1 $\pm$ 0.1	65.9 $\pm$ 0.1
Hierarchical (8192 + LWAN + L*)	$\mathbb{T}$	91.9 $\pm$ 0.2	94.1 $\pm$ 0.2	65.5 $\pm$ 0.7	71.1 $\pm$ 0.4	66.4 $\pm$ 0.3

Table 2: Comparison of TrLDC against state-of-the-art on the MIMIC-III test set. $\mathbb{C}$: CNN-based models; $\mathbb{R}$: RNN-based models; and $\mathbb{T}$: Transformer-based models. Models marked with an asterisk (*) is domain-specific RoBERTa-Large (Lewis et al., 2020), whereas Longformer and other RoBERTa models are task-adaptive pre-trained base versions.

	ECtHR	20 News	Hyper
First (512)	73.5 $\pm$ 0.2	86.1 $\pm$ 0.3	92.9 $\pm$ 3.2
Random (512)	79.0 $\pm$ 0.6	85.3 $\pm$ 0.4	88.9 $\pm$ 2.5
Informative (512)	72.4 $\pm$ 0.2	86.2 $\pm$ 0.3	91.7 $\pm$ 3.2
Longformer (4096)	81.0 $\pm$ 0.5	**86.3** $\pm$ **0.5**	**97.9** $\pm$ **0.7**
Hierarchical (4096)	**81.1** $\pm$ **0.2**	**86.3** $\pm$ **0.2**	95.4 $\pm$ 1.3

Table 3: Comparison of TrLDC against baselines processing up to 512 tokens. We report Micro F_1 on ECtHR, Accuracy on 20 News and Hyperpartisan datasets.

best performing CNN-based model (Ji et al., 2021b) on MIMIC-III. By processing longer text and using the RoBERTa-Large model, the hierarchical models further improve the performance, leading to comparable results of RNN-based models (Vu et al., 2020; Yuan et al., 2022). We hypothesise that further improvements can be observed when TrLDC models are enhanced with better hierarchy-aware classifier as in Vu et al. (2020) or code synonyms are used for training as in Yuan et al. (2022).

6 Practical Advice

We compile several questions that practitioners may ask regarding long document classification and provide answers based on our results:

Q1 When should I start to consider using long document classification models?

A We suggest using TrLDC models if you work with datasets consisting of long documents (e.g., 2K tokens on average). We notice that on 20 News dataset, the gap between baselines that process 512 tokens and long document models is negligible.[9]

Q2 Which model should I choose? Longformer or hierarchical Transformers?

A We suggest Longformer as the starting point if you do not plan on extensively tuning hyperparameters. We find the default config of Longformer is robust, although it is possible to set a moderate size (64-128) of local attention window to improve efficiency without sacrificing effectiveness, and a small number of additional global attention tokens to make the training more stable. On the other hand, hierarchical Transformers may benefit from careful hyperparameter tuning (e.g., document splitting strategy, using LWAN). We suggest splitting a document into small non-structure-derived segments

[9] Although Hyperpartisan is a widely used benchmark for long document models, we do not recommend drawing practical conclusions based on our results because we observe high variance when we run experiments using different GPUs or CUDA versions. We attribute this may to the small size (65) of its test set and the subjectivity of the task.

(e.g., 128 tokens) which overlap as a starting point when employing hierarchical Transformers.

We also note that the publicly available Longformer models can process sequences up-to 4096 tokens, whereas hierarchical Transformers can be easily extended to process much longer sequence.

7 Related Work

Long document classification Document length was not a point of controversy in the pre-neural era of NLP, where documents are encoded with Bag-of-Word representations, e.g., TF-IDF scores. The issue arised with the introduction of deep neural networks. Tang et al. (2015) use CNN and BiLSTM based hierarchical networks in a bottom-up fashion, i.e., first encode sentences into vectors, then combine those vectors in a single document vector. Similarly, Yang et al. (2016) incorporate the attention mechanism when constructing the sentence and document representation. Hierarchical variants of BERT have also been explored for document classification (Mulyar et al., 2019; Chalkidis et al., 2022), abstractive summarization (Zhang et al., 2019), semantic matching (Yang et al., 2020). Both Zhang et al., and Yang et al. also propose specialised pre-training tasks to explicitly capture sentence relations within a document. A very recent work by Park et al. (2022) shows that TrLDC do not perform consistently well across datasets that consist of 700 tokens on average.

Methods of modifying transformer architecture for long documents can be categorised into two approaches: *recurrent* Transformers and *sparse* attention Transformers. The recurrent approach processes segments moving from left-to-right (Dai et al., 2019). To capture bidirectional context, Ding et al. (2021) propose a retrospective mechanism in which segments from a document are fed twice as input. Sparse attention Transformers have been explored to reduce the complexity of self-attention, via using dilated sliding window (Child et al., 2019), and locality-sensitive hashing attention (Kitaev et al., 2020). Recently, the combination of local (window) and global attention are proposed by Beltagy et al. (2020) and Zaheer et al. (2020), which we have detailed in Section 3.

ICD Coding The task of assigning most relevant ICD codes to a document, e.g., radiology report (Pestian et al., 2007), death certificate (Koopman et al., 2015) or discharge summary (Johnson et al., 2016), as a whole, has a long history of development (Farkas and Szarvas, 2008). Most existing methods simplified this task as a text classification problem and built classifiers using CNNs (Karimi et al., 2017) or LSTMs (Xie et al., 2018). Since the number of unique ICD codes is very large, methods are proposed to exploit relation between codes based on label co-occurrence (Dong et al., 2021), label count (Du et al., 2019), knowledge graph (Xie et al., 2019; Cao et al., 2020; Lu et al., 2020), code's textual descriptions (Mullenbach et al., 2018; Rios and Kavuluru, 2018). More recently, Ji et al. (2021a); Gao et al. (2021) investigate various methods of applying BERT on ICD coding. Different from our work, they mainly focus on comparing domain-specific BERT models that are pre-trained on various types of corpora. Ji et al. show that PubMedBERT—pre-trained from scratch on PubMed abstracts—outperforms other variants pre-trained on clinical notes or health-related posts; Gao et al. show that BlueBERT—pre-trained on PubMed and clinical notes—performs best. However, both report that Transformers-based models perform worse than CNN-based ones.

8 Conclusions

Transformers have previously been criticised for being incapable of long document classification. In this paper, we carefully study the role of different components of Transformer-based long document classification models. By conducting experiments on MIMIC-III and other three datasets (i.e., ECtHR, 20 News and Hyperpartisan), we observe clear improvements in performance when a model is able to process more text. Firstly, Longformer, a sparse attention model, which can process up to 4096 tokens, achieves competitive results with CNN-based models on MIMIC-III; its performance is relatively robust; a moderate size of local attention window (e.g., 128) and a small number (e.g., 16) of evenly chosen tokens with global attention can improve the efficiency and stability without sacrificing its effectiveness. Secondly, hierarchical Transformers outperform all CNN-based models by a large margin; the key design choice is how to split a document into segments which can be encoded by pre-trained models; although the best performing segment length is dataset dependent, we find splitting a document into small overlapping segments (e.g., 128 tokens) is an effective strategy. Taken together, these experiments rebut the criticisms of Transformers for long document classification.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. DocBERT: BERT for Document Classification. *arXiv*, 1904.08398.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv*, 2004.05150.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. HyperCore: Hyperbolic and Co-graph Representation for Automatic ICD Coding. In *ACL*.

Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. An Empirical Study on Large-Scale Multi-Label Text Classification Including Few and Zero-Shot Labels. In *EMNLP*.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael J Bommarito II, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. In *ACL*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv*, 1904.10509.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking Attention with Performers. In *ICLR*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

SiYu Ding, Junyuan Shang, Shuohuan Wang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE-Doc: A Retrospective Long-Document Modeling Transformer. In *ACL-IJCNLP*.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus. In *EMNLP*.

Hang Dong, Víctor Suárez-Paniagua, William Whiteley, and Honghan Wu. 2021. Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *JBI*, 116.

Jingcheng Du, Qingyu Chen, Yifan Peng, Yang Xiang, Cui Tao, and Zhiyong Lu. 2019. ML-Net: multi-label classification of biomedical texts with deep neural networks. *JAMIA*, 26.

Richárd Farkas and György Szarvas. 2008. Automatic construction of rule-based ICD-9-CM coding systems. *BMC Bioinform.*, 9.

Shang Gao, Mohammed Alawad, M. Todd Young, John Gounley, Noah Schaefferkoetter, Hong Jun Yoon, Xiao-Cheng Wu, Eric B. Durbin, Jennifer Doherty, Antoinette Stroup, Linda Coyle, and Georgia Tourassi. 2021. Limitations of Transformers on Clinical Text Classification. *IEEE J. Biomed. Health Inform.*, 25.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL*.

Shaoxiong Ji, Matti Hölttä, and Pekka Marttinen. 2021a. Does the Magic of BERT Apply to Medical Code Assignment? A Quantitative Study. *Comput. Biol. Med.*, 139.

Shaoxiong Ji, Shirui Pan, and Pekka Marttinen. 2021b. Medical Code Assignment with Gated Convolution and Note-Code Interaction. In *Findings of ACL-IJCNLP*.

Thorsten Joachims. 1997. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In *ICML*.

Alistair E W Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Sci. Data*, 3.

Sarvnaz Karimi, Xiang Dai, Hamed Hassanzadeh, and Anthony Nguyen. 2017. Automatic Diagnosis Coding of Radiology Reports: A Comparison of Deep Learning and Conventional Classification Methods. In *BioNLP@ACL*.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 Task 4: Hyperpartisan News Detection. In *SemEval@NAACL*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *ICLR*.

Bevan Koopman, Sarvnaz Karimi, Anthony Nguyen, Rhydwyn McGuire, David Muscatello, Madonna Kemp, Donna Truran, Ming Zhang, and Sarah Thackway. 2015. Automatic classification of diseases from free-text death certificates for real-time surveillance. *BMC Medical Inform. Decis. Mak.*, 15.

Rasmus Kær Jørgensen, Mareike Hartmann, Xiang Dai, and Desmond Elliott. 2021. mDAPT: Multilingual Domain Adaptive Pretraining in a Single Model. In *Findings of EMNLP*.

Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. 2020. Pretrained Language Models for Biomedical and Clinical Tasks: Understanding and Extending the State-of-the-Art. In *ClinicalNLP@EMNLP*.

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*, 1907.11692.

Jueqing Lu, Lan Du, Ming Liu, and Joanna Dipnall. 2020. Multi-label Few/Zero-shot Learning with Knowledge Aggregated from Multiple Label Graphs. In *EMNLP*.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines. In *ICLR*.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable Prediction of Medical Codes from Clinical Text. In *NAACL*.

Andriy Mulyar, Elliot Schumacher, Masoud Rouhizadeh, and Mark Dredze. 2019. Phenotyping of Clinical Notes with Improved Document Classification Models Using Contextualized Neural Language Models. *arXiv*, 1910.13664.

Hyunji Park, Yogarshi Vyas, and Kashif Shah. 2022. Efficient Classification of Long Documents Using Transformers. In *ACL*.

Damian Pascual, Sandro Luck, and Roger Wattenhofer. 2021. Towards BERT-based Automatic ICD Coding: Limitations and Opportunities. In *BioNLP@NAACL*.

John Pestian, Chris Brew, Pawel Matykiewicz, Dj J Hovermale, Neil Johnson, K Bretonnel Cohen, and Wlodzislaw Duch. 2007. A shared task involving multi-label classification of clinical free text. In *BioNLP@ACL*.

Jiezhong Qiu, Hao Ma, Omer Levy, Wen-tau Yih, Sinong Wang, and Jie Tang. 2020. Blockwise Self-Attention for Long Document Understanding. In *Findings of EMNLP*.

Alan Ramponi and Barbara Plank. 2020. Neural Unsupervised Domain Adaptation in NLP—A Survey. In *COLING*.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces. In *EMNLP*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification? In *CCL*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient Transformers: A Survey. *arXiv*, 2009.06732.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for ICD coding from clinical text. In *IJCAI*.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-Specific Document Representation for Multi-Label Text Classification. In *EMNLP-IJCNLP*.

Pengtao Xie, Haoran Shi, Ming Zhang, and Eric Xing. 2018. A Neural Architecture for Automated ICD Coding. In *ACL*.

Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. 2019. EHR Coding with Multi-scale Feature Attention and Structured Knowledge Graph Propagation. In *CIKM*.

Wenhan Xiong, Barlas Oğuz, Anchit Gupta, Xilun Chen, Diana Liskovich, Omer Levy, Wen-tau Yih, and Yashar Mehdad. 2021. Simple Local Attentions Remain Competitive for Long-Context Tasks. *arXiv*, 2112.07210.

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

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *NAACL*.

Zheng Yuan, Chuanqi Tan, and Songfang Huang. 2022. Code Synonyms Do Matter: Multiple Synonyms Matching Network for Automatic ICD Coding. In *ACL*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, and Li Yang. 2020. Big Bird: Transformers for Longer Sequences. In *NeurIPS*.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization. In *ACL*.

9 Appendix

9.1 Limitations

Long document classification datasets are usually annotated using a large number of labels. For example, the complete MIMIC-III dataset contains $8,692$ unique labels. As we mentioned in Section 2, we focus on building document representation and leave the challenge of learning with a *large target label* set for future work. Therefore, in this paper, we follow previous work (Mullenbach et al., 2018; Chalkidis et al., 2022) and consider a subset of frequent labels in MIMIC-III and ECtHR.

9.2 Dataset statistics

Table 4 shows the descriptive statistics of four datasets we use.

	Train	Dev	Test
MIMIC-III			
Documents	8,066	1,573	1,729
Unique labels	50	50	50
Avg. tokens	2,260	2,693	2,737
ECtHR			
Documents	8,866	973	986
Unique labels	10	10	10
Avg. tokens	2,140	2,345	2,532
Hyperpartisan			
Documents	516	64	65
Unique labels	2	2	2
Avg. tokens	741	707	845
20 News			
Documents	10,183	1,131	7,532
Unique labels	20	20	20
Avg. tokens	613	627	551

Table 4: Statistics of the datasets. The number of tokens is calculated using RoBERTa tokenizer.

9.3 An illustration of sparse attention

Figure 7 shows a comparison of three types of attention operations. Longformer uses the combination of local attention and global attention.

9.4 Details of task-adaptive pre-training

Hyperparameters and training time for task-adaptive pre-training can be found in Table 5.

9.5 Details of classification experiments

Preprocessing We mainly follow Mullenbach et al. (2018) to preprocess the MIMIC-III dataset.

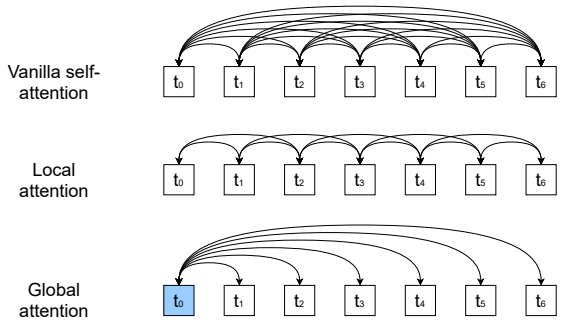


Figure 7: A comparison of three types of attention operations. The example sequence contains 7 tokens; we set local attention window size as 2, and only the first token using global attention. Note that these curves are bi-directional that tokens can attend to each other.

	Longformer	RoBERTa
Max sequence	4096	128
Batch size	8	128
Learning rate	5e-5	5e-5
Training epochs	6	15
Training time (GPU-hours)	≈ 130	≈ 40

Table 5: Hyperparameters and training time (measured on MIMIC-III dataset) for task-adaptive pre-training Longformer and RoBERTa. Batch size = batch size per GPU $\times$ num. GPUs $\times$ gradient accumulation steps.

That is, we lowercase the text, remove all punctuation marks and tokenize text by white spaces. The only change we make is that we normalise numeric (e.g., convert '2021' to '0000') instead of deleting numeric-only tokens in Mullenbach et al. (2018). We did not apply additional preprocessing to ECtHR and 20 News. We follow Beltagy et al. (2020) to preprocess the Hyperpartisan dataset.[10]

Training We fine-tune the multilabel classification model using a binary cross entropy loss. That is, given an training example whose ground truth and predicted probability for the i-th label are y_i (0 or 1) and $\hat{y}_i$, we calculate its loss, over the C unique classification labels, as:

$$\mathcal{L} = \sum_{i=1}^{C} -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i).$$

For the multiclass and binary classification tasks, we fine-tune using the cross entropy loss, where $\hat{y}_g$ is the predicted probability for the gold label:

$$\mathcal{L} = -\log(\hat{y}_g),$$

[10] https://github.com/allenai/longformer

We use the same effective batch size (16), learning rate (2e-5), maximum number of training epochs (30) with early stop patience (5) in all experiments. We also follow Longformer (Beltagy et al., 2020) and set the maximum sequence length as 4096 in most of the experiments unless other specified. We fine-tune all classification models on Quadro RTX 6000 (24 GB GPU memory) or Tesla V100 (32 GB GPU memory). If one batch of data is too large to fit into the GPU memory, we use gradient accumulation so that the effective batch sizes (batch size per GPU $\times$ gradient accumulation steps) are still the same.

We repeat all experiments five times with different random seeds. The model which is most effective on the development set, measured using the micro F_1 score (multilabel) or accuracy (multiclass and binary), is used for the final evaluation.

9.6 A comparison between clinical notes and legal cases

Although we usually use the term *domain* to indicate that texts talk about a narrow set of related concepts (e.g., clinical concepts or legal concepts), text can vary along different dimensions (Ramponi and Plank, 2020).

In addition to the statistics difference between MIMIC-III and ECtHR, which we show in Table 4, there is another difference worthy considering: clinical notes are private as they contain protected health information. Even those clinical notes after de-identification are usually not publicly available (e.g., downloadable using web crawler). In contrast, legal cases have generally been allowed and encouraged to share with the public, and thus become a large portion of crawled pre-training data (Dodge et al., 2021). Dodge et al. find that legal documents, especially U.S. case law, are a significant part of the C4 corpus, a cleansed version of CommonCrawl used to pre-train RoBERTa models. The ECtHR proceedings are also publicly available via HUDOC, the court's database.

We suspect task-adaptive pre-training is more useful on MIMIC-III than on ECtHR (Figure 4) may relate to this difference. Therefore, we evaluate the vanilla RoBERTa on MIMIC-III and ECtHR regarding tokenization and language modelling. A comparison of the fragmentation ratio using the tokenizer and perplexity using the language model can be found in Table 6.

	MIMIC-III	ECtHR
Fragmentation ratio	1.233	1.118
Perplexity	1.351	1.079

Table 6: Evaluating vanilla RoBERTa on MIMIC-III and ECtHR. Lower fragmentation ratio and perplexity indicate that the test data have a higher similarity with the RoBERTa pre-training data.

9.7 A comparison between TAPT and public available RoBERTa by (Lewis et al., 2020)

We compare our TAPT-RoBERTa against publicly available domain-specific RoBERTa (Lewis et al., 2020), which are trained from scratch on biomedical articles and clinical notes, in hierarchical models. In these experiments, we split long documents into overlapping segments of 64 tokens. Results in Figure 8 show that TAPT-RoBERTa outperforms domain-specific base model, but underperforms the larger model.

9.8 Results on ECtHR test set

Results in Table 7 show that our results are higher than the ones reported in (Chalkidis et al., 2022). Chalkidis et al. compare different BERT variants including domain-specific models, whereas we use task-adaptive pre-trained models. Regarding hierarchical method, we split a document into overlapping segments, each of which has 512 tokens. We use the default setting for Longformer as in Beltagy et al. (2020).

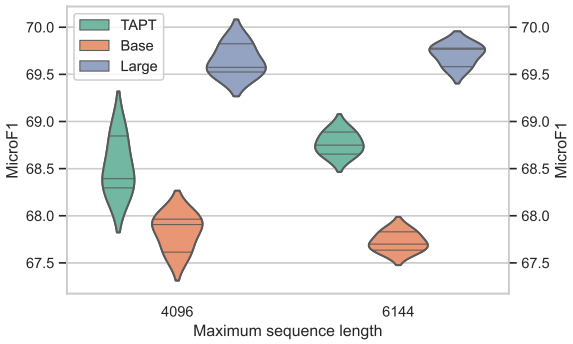


Figure 8: A comparison of task-adaptive pre-trained RoBERTa against public available domain-specific RoBERTa. Both Base and Large RoBERTa models are trained from scratch on biomedical articles and clinical notes (Lewis et al., 2020).

	Macro F_1	Micro F_1
RoBERTa	68.9	77.3
CaseLaw-BERT	70.3	78.8
BigBird	70.9	78.8
DeBERTa	71.0	78.8
Longformer	71.7	79.4
BERT	73.4	79.7
Legal-BERT	74.7	80.4
Longformer (4096)	76.0 $\pm$ 1.4	80.7 $\pm$ 0.3
Hierarchical (4096)	76.6 $\pm$ 0.7	81.0 $\pm$ 0.3

Table 7: Comparison of our results against the results reported in (Chalkidis et al., 2022) on the ECtHR test set. Results are sorted by Micro F_1.

9.9 A comparison between Longformer and Hierarchical model

Table 8 shows a comparison between Longformer and Hierarchical models regarding the number of parameters and their GPU consumption. We use batch size of 2 in these experiments, and measure the impact of attention window size and segment length on the memory footprint.

Size	Longformer (148.6M)	Hierarchical (139.0M)
Maximum sequence length: 1024		
64	4.8G	3.6G
128	5.0G	3.8G
256	5.5G	4.1G
512	6.6G	4.7G
Maximum sequence length: 4096		
64	11.8G	7.8G
128	12.8G	8.4G
256	14.9G	9.6G
512	19.4G	12.2G

Table 8: A comparison between Longformer and Hierarchical models. The number of parameters are listed in the table header. Size refers to the local attention window size in Longformer and the segment length in hierarchical method, respectively.

9.10 A comparison between evenly splitting and splitting based on document structure

Figure 9 shows that splitting by the paragraph level document structure does not improve performance on the ECtHR dataset. On MIMIC-III, splitting based on document structure substantially underperforms evenly splitting the document.

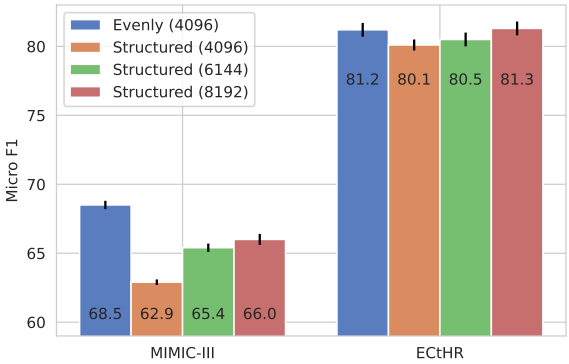


Figure 9: A comparison between evenly splitting and splitting based on document structure.

9.11 Detailed results on the development sets

For the sake of brevity, we use only micro F_1 score in most of our illustrations, and we detail results of other metrics in this section.

Max sequence length	AUC		F_1		P@5
	Macro	Micro	Macro	Micro	
512	81.4 ± 0.1	85.1 ± 0.2	39.2 ± 0.9	52.2 ± 0.3	53.3 ± 0.3
1024	83.6 ± 0.2	87.3 ± 0.3	43.2 ± 0.6	56.3 ± 0.5	56.5 ± 0.2
2048	86.5 ± 0.2	89.8 ± 0.1	48.2 ± 1.1	60.5 ± 0.4	59.4 ± 0.3
4096	88.4 ± 0.1	91.5 ± 0.1	53.1 ± 0.5	64.0 ± 0.3	62.0 ± 0.4

Table 9: Detailed results of Figure 1: the effectiveness of Longformer on the MIMIC-III development set.

	AUC		F_1		P@5
	Macro	Micro	Macro	Micro	
Longformer on **MIMIC-III**					
Vanilla	88.4 ± 0.1	91.5 ± 0.1	53.1 ± 0.5	64.0 ± 0.3	62.0 ± 0.4
TAPT	90.3 ± 0.2	92.7 ± 0.1	60.8 ± 0.4	68.5 ± 0.3	64.8 ± 0.3
RoBERTa on **MIMIC-III**					
Vanilla	81.6 ± 0.2	85.0 ± 0.3	43.2 ± 1.7	53.9 ± 0.4	54.0 ± 0.2
TAPT	82.3 ± 0.4	85.5 ± 0.3	48.8 ± 0.4	56.7 ± 0.2	55.3 ± 0.2
Longformer on **ECtHR**					
Vanilla	—	—	77.4 ± 2.3	81.3 ± 0.3	—
TAPT	—	—	78.5 ± 2.2	82.1 ± 0.6	—
RoBERTa on **ECtHR**					
Vanilla	—	—	72.2 ± 1.5	74.8 ± 0.4	—
TAPT	—	—	72.7 ± 0.7	75.1 ± 0.4	—

Table 10: Detailed results of Figure 4: the impact of task-adaptive pre-training. Note that we use maximum sequence length 512 for RoBERTa and 4096 for Longformer in these experiments.

	AUC		F_1			
Size	Macro	Micro	Macro	Micro	P@5	Accuracy
MIMIC-III						
32	89.8 ± 0.1	92.3 ± 0.1	59.6 ± 0.6	67.9 ± 0.3	64.2 ± 0.3	—
64	90.0 ± 0.1	92.5 ± 0.1	60.3 ± 0.3	68.1 ± 0.1	64.5 ± 0.1	—
128	90.1 ± 0.1	92.6 ± 0.1	60.5 ± 0.7	68.3 ± 0.3	64.7 ± 0.3	—
256	90.2 ± 0.0	92.6 ± 0.1	60.7 ± 0.6	68.4 ± 0.3	64.6 ± 0.2	—
512	90.3 ± 0.2	92.7 ± 0.1	60.8 ± 0.4	68.5 ± 0.3	64.8 ± 0.3	—
ECtHR						
32	—	—	78.2 ± 1.2	81.2 ± 0.3	—	—
64	—	—	78.6 ± 1.7	81.4 ± 0.1	—	—
128	—	—	79.9 ± 1.6	82.1 ± 0.5	—	—
256	—	—	78.5 ± 2.1	81.8 ± 0.4	—	—
512	—	—	78.5 ± 2.2	82.1 ± 0.6	—	—
Hyperpartisan						
32	—	—	–	–	—	83.9 ± 0.7
64	—	—	–	–	—	83.3 ± 1.9
128	—	—	–	–	—	83.9 ± 0.7
256	—	—	–	–	—	88.0 ± 0.7
512	—	—	–	–	—	85.9 ± 2.2
20 News						
32	—	—	–	–	—	92.8 ± 0.6
64	—	—	–	–	—	94.0 ± 0.5
128	—	—	–	–	—	93.8 ± 0.3
256	—	—	–	–	—	93.5 ± 0.1
512	—	—	–	–	—	94.0 ± 0.1

Table 11: The impact of local attention window size in Longformer, measured on the development sets.

	AUC		F_1		
# tokens	Macro	Micro	Macro	Micro	P@5
MIMIC-III					
1	90.1 ± 0.2	92.6 ± 0.1	60.5 ± 0.9	68.2 ± 0.3	64.7 ± 0.3
8	90.0 ± 0.1	92.5 ± 0.1	60.5 ± 0.7	68.2 ± 0.3	64.6 ± 0.2
16	90.0 ± 0.2	92.5 ± 0.1	60.0 ± 0.2	68.1 ± 0.2	64.3 ± 0.3
32	90.0 ± 0.2	92.4 ± 0.1	60.1 ± 0.5	67.9 ± 0.1	64.4 ± 0.2
64	89.9 ± 0.2	92.4 ± 0.1	59.9 ± 1.0	67.9 ± 0.4	64.4 ± 0.3
ECtHR					
1	—	—	78.5 ± 1.8	80.8 ± 0.4	—
8	—	—	77.2 ± 2.0	80.8 ± 0.4	—
16	—	—	77.7 ± 0.4	80.7 ± 0.3	—
32	—	—	78.2 ± 1.4	80.6 ± 0.4	—
64	—	—	77.7 ± 2.3	80.7 ± 0.5	—

Table 12: Detailed results of Figure 5: the effect of applying global attention on more tokens, which are evenly chosen based on their positions.

	AUC		F_1		
# tokens	Macro	Micro	Macro	Micro	P@5
MIMIC-III					
1	90.1 $\pm$ 0.2	92.6 $\pm$ 0.1	60.5 $\pm$ 0.9	68.2 $\pm$ 0.3	64.7 $\pm$ 0.3
8	89.7 $\pm$ 0.2	92.0 $\pm$ 0.1	61.0 $\pm$ 1.3	66.9 $\pm$ 0.4	64.0 $\pm$ 0.4
16	89.4 $\pm$ 0.2	91.9 $\pm$ 0.1	60.1 $\pm$ 1.2	66.5 $\pm$ 0.3	63.9 $\pm$ 0.5
32	89.4 $\pm$ 0.4	91.9 $\pm$ 0.2	60.3 $\pm$ 1.6	66.4 $\pm$ 0.6	63.7 $\pm$ 0.7
64	89.1 $\pm$ 0.4	91.7 $\pm$ 0.2	59.4 $\pm$ 2.0	66.2 $\pm$ 0.7	63.4 $\pm$ 0.7
ECtHR					
1	—	—	78.5 $\pm$ 1.8	80.8 $\pm$ 0.4	—
8	—	—	79.2 $\pm$ 0.3	80.9 $\pm$ 0.2	—
16	—	—	77.6 $\pm$ 1.2	80.4 $\pm$ 0.4	—
32	—	—	77.1 $\pm$ 0.7	80.0 $\pm$ 0.2	—
64	—	—	76.6 $\pm$ 1.1	79.9 $\pm$ 0.5	—

Table 13: The effect of applying global attention on more informative tokens, which are identified based on TF-IDF.

	AUC		F_1			
Size	Macro	Micro	Macro	Micro	P@5	Accuracy
Disjoint segments on **MIMIC-III**						
64	89.4 ± 0.1	92.0 ± 0.1	60.8 ± 1.1	67.9 ± 0.3	63.5 ± 0.3	—
128	89.5 ± 0.1	92.1 ± 0.1	61.2 ± 0.6	68.0 ± 0.3	63.5 ± 0.3	—
256	89.6 ± 0.1	92.1 ± 0.1	61.0 ± 0.4	67.6 ± 0.2	63.6 ± 0.2	—
512	89.2 ± 0.2	91.8 ± 0.2	59.4 ± 0.5	66.7 ± 0.3	63.4 ± 0.4	—
Overlapping segments on **MIMIC-III**						
64	89.7 ± 0.1	92.3 ± 0.1	62.3 ± 0.2	68.7 ± 0.1	64.1 ± 0.1	—
128	89.7 ± 0.2	92.3 ± 0.1	61.8 ± 0.9	68.5 ± 0.3	64.0 ± 0.2	—
256	89.5 ± 0.1	92.1 ± 0.1	61.4 ± 0.3	68.1 ± 0.2	63.8 ± 0.1	—
512	89.4 ± 0.1	92.0 ± 0.0	60.3 ± 0.3	67.2 ± 0.2	63.6 ± 0.3	—
Disjoint segments on **ECtHR**						
64	—	—	76.6 ± 1.2	79.7 ± 0.2	—	—
128	—	—	77.6 ± 2.3	80.8 ± 0.4	—	—
256	—	—	77.7 ± 1.4	81.2 ± 0.4	—	—
512	—	—	78.3 ± 1.3	81.7 ± 0.3	—	—
Overlapping segments on **ECtHR**						
64	—	—	76.9 ± 1.7	80.5 ± 0.5	—	—
128	—	—	77.5 ± 1.7	81.2 ± 0.5	—	—
256	—	—	78.1 ± 1.4	81.5 ± 0.2	—	—
512	—	—	78.4 ± 1.5	81.4 ± 0.4	—	—
Disjoint segments on **Hyperpartisan**						
64	—	—	—	—	—	88.8 ± 1.8
128	—	—	—	—	—	89.1 ± 1.4
256	—	—	—	—	—	87.8 ± 1.8
512	—	—	—	—	—	86.2 ± 1.8
Overlapping segments on **Hyperpartisan**						
64	—	—	—	—	—	87.5 ± 1.4
128	—	—	—	—	—	88.4 ± 1.2
256	—	—	—	—	—	88.1 ± 2.1
512	—	—	—	—	—	88.4 ± 0.8
Disjoint segments on **20 News**						
64	—	—	—	—	—	93.3 ± 0.2
128	—	—	—	—	—	93.5 ± 0.3
256	—	—	—	—	—	94.4 ± 0.4
512	—	—	—	—	—	94.0 ± 0.3
Overlapping segments on **20 News**						
64	—	—	—	—	—	93.8 ± 0.4
128	—	—	—	—	—	93.4 ± 0.3
256	—	—	—	—	—	94.5 ± 0.2
512	—	—	—	—	—	93.9 ± 0.3

Table 14: The effect of varying the segment length and whether allowing segments to overlap in the hierarchical transformers.

	AUC		F_1		
	Macro	Micro	Macro	Micro	P@5
MIMIC-III					
E (4096)	89.7 ± 0.2	92.3 ± 0.1	61.8 ± 0.9	68.5 ± 0.3	64.0 ± 0.2
S (4096)	87.2 ± 0.2	90.1 ± 0.2	55.2 ± 0.4	62.9 ± 0.2	59.9 ± 0.2
S (6144)	88.2 ± 0.2	91.0 ± 0.2	57.8 ± 0.3	65.4 ± 0.3	61.7 ± 0.3
S (8192)	88.5 ± 0.3	91.2 ± 0.2	58.8 ± 0.2	66.0 ± 0.4	62.4 ± 0.1
ECtHR					
E (4096)	—	—	77.5 ± 1.7	81.2 ± 0.5	—
S (4096)	—	—	75.3 ± 1.3	80.1 ± 0.4	—
S (6144)	—	—	77.1 ± 1.8	80.5 ± 0.5	—
S (8192)	—	—	77.7 ± 1.9	81.3 ± 0.5	—

Table 15: Detailed results of Figure 9: a comparison between evenly splitting and splitting based on document structure. E: evenly splitting; S: splitting based on document structure.

	AUC		F_1		
	Macro	Micro	Macro	Micro	P@5
MIMIC-III					
Longformer	90.0 ± 0.2	92.5 ± 0.1	60.0 ± 0.2	68.1 ± 0.2	64.3 ± 0.3
+ LWAN	90.5 ± 0.2	92.9 ± 0.2	62.2 ± 0.7	69.2 ± 0.3	65.1 ± 0.1
Hierarchical	89.7 ± 0.2	92.3 ± 0.1	61.8 ± 0.9	68.5 ± 0.3	64.0 ± 0.2
+ LWAN	91.4 ± 0.1	93.7 ± 0.1	64.2 ± 0.4	70.3 ± 0.1	65.3 ± 0.1
ECtHR					
Longformer	—	—	77.7 ± 0.4	80.7 ± 0.3	—
+ LWAN	—	—	79.5 ± 0.8	81.1 ± 0.3	—
Hierarchical	—	—	77.5 ± 1.7	81.2 ± 0.5	—
+ LWAN	—	—	79.7 ± 0.9	81.3 ± 0.3	—

Table 16: The effect of label-wise attention network.