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 long document classification 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 attention window, use of global attention) and hierarchical based (e.g., document splitting strategy) transformers on two different datasets, and we derive practical advice of applying Transformer-based models on long document classification tasks. We find that, if applied properly, Transformer-based models can outperform former state-of-the-art CNN based models on MIMIC-III, a challenging dataset from the clinical domain.

1 Introduction

The pre-train–fine-tune paradigm has become the de-facto practice since the introduction of BERT (Devlin et al., 2019; Liu et al., 2019). However, the recent literature in text classification mostly focuses on short sequences, such as sentences or paragraphs (Sun et al., 2019; Wei and Zou, 2019; Mosbach et al., 2021), which are sometimes misleadingly named as documents,[1] a term commonly used to denote an article or even a book.

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 hardly a long document. A common practice is to truncate 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 a very naive approach for long document classification because truncating the text may omit important information, leading to poor classification performance. See Figure 1 for empirical evidence to support this claim. Another challenge is the computational foot-print of conventional Transformer-based models: in the standard 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.

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

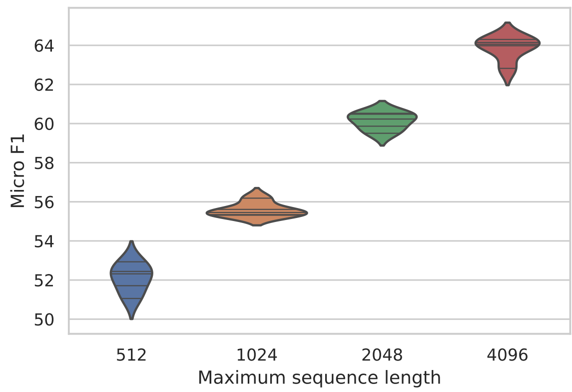


Figure 1: The effectiveness of Longformer, a long-document Transformer, on the MIMIC-III development set. There is a clear benefit from being able to process longer text.

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

MIMIC-III dataset (Johnson et al., 2016) with an 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., 2021; Gao et al., 2021; Pascual et al., 2021). There is a clear need to understand the performance of Transformer-based models on documents that are actually long.

In this work, we 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, if applied properly, Transformer-based models can outperform former state-of-the-art CNN based models on MIMIC-III.

- We conduct careful analyses to understand the impact of several design choices on both the effectiveness and efficiency of different approaches. Based on our empirical results on two challenging datasets from clinical and legal domains, we derive practical advice of applying Transformer-based models to long document classification.

2 Problem Formulation and Datasets

We divide the document classification model into two components: (1) a document encoder, which builds 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 importance of 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 standard multi-label classifier, i.e., a linear layer with C outputs, where C is the number of classes, followed by sigmoid activations, trained using binary cross entropy loss.[2]

We use two datasets—MIMIC-III (Johnson et al., 2016) and ECtHR (Chalkidis et al., 2021)—from clinical and legal domains respectively. The statistics of the datasets can be found in Table 1 and the document length distribution is shown in Figure 2.

	Train	Dev	Test
MIMIC-III			
Documents	8,066	1,573	1,729
Unique labels	50	50	50
Avg. words	1,833	2,177	2,210
Avg. subtokens	2,260	2,693	2,737
90th pctl. subtokens	3,757	4,078	4,216
ECtHR			
Documents	8,866	973	986
Unique labels	10	10	10
Avg. words	1,914	2,125	2,284
Avg. subtokens	2,140	2,345	2,532
90th pctl. subtokens	4,762	4,930	5,576

Table 1: Statistics of the datasets. The number of words and subtokens is calculated using RoBERTa tokenizer.

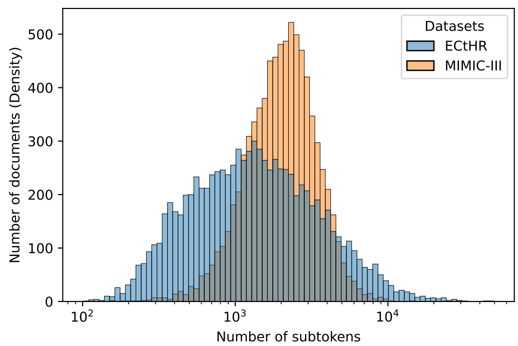


Figure 2: The distribution of document lengths. A log-10 scale is used for the X axis.

MIMIC-III contains approx. 50K discharge summaries from a US hospital. Each summary 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]

The ECtHR dataset contains 11K 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. Each case is mapped to one or more *articles* of the convention that were *allegedly* violated (considered by the court).[4]

[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.

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

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

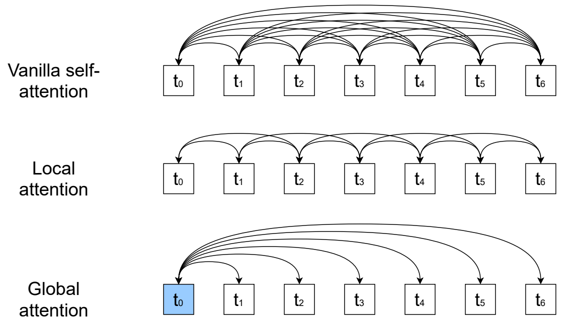


Figure 3: A comparison of three types of attention operations. The example sequence contains 7 tokens; we set local attention window size as 2, and only the first token using global attention. Note that these curves are bi-directional that tokens can attend to each other.

3 Approaches

In the era of Transformer-based models, we identify two approaches in the literature that aim to mitigate the computational complexity of the original transformer: *sparse*, and *hierarchical* Transformers.

3.1 Sparse-Attention Transformers

Vanilla transformers rely 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. Several Transformer-based models (Kitaev et al., 2020; Choromanski et al., 2021) have been proposed exploring *sparse attention* alternatives that scale linearly, thus it can be used to process long sequences.

Longformer of Beltagy et al. (2020) extends Transformer-based models to support longer sequences, using sparse-attention. It consists of local (window-based) attention and global attention that reduces the computational complexity of the model and thus can be deployed to process longer text (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 3). **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.

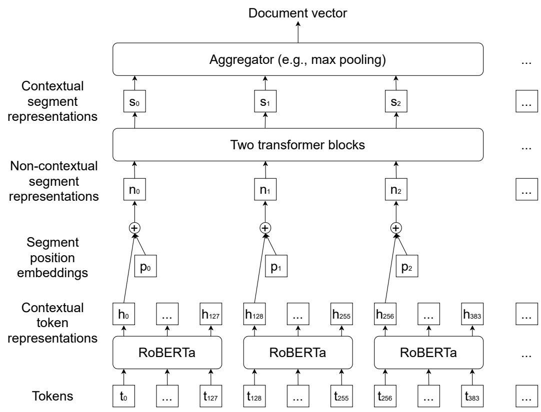


Figure 4: A high-level illustration of hierarchical Transformers. A shared pre-trained RoBERTa is used to encode each segment, and a two layer transformer blocks is used to capture the interaction between different segments. Finally, contextual segment representations are aggregated into a document representation.

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 4). We sum the contextual representation of the first token ([CLS]) from each segment up with segment position embeddings—sinusoidal initialised (Vaswani et al., 2017) and keep trainable—as the segment representation (i.e., n_i in Figure 4). 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 4), which are finally aggregated into a document representation. By default, the aggregator is the **max-pooling operation** unless other specified.[5]

4 Experimental Setup

Backbone Models We consider two models: Longformer (Beltagy et al., 2020), and RoBERTa-based (Liu et al., 2019) hierarchical Transformers.

Evaluation metrics For the MIMIC-III dataset, we follow previous work (Mullenbach et al., 2018;

[5]Code is available at [ANON].

Cao et al., 2020) and use micro-averaged AUC (Area Under the receiver 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. For the ECtHR dataset, we use both micro and macro averaged F_1. For the sake of brevity, we use micro F_1 score as the main metric in most of our illustrations, and results of other metrics are detailed in the Appendix.

Preprocessing We mainly follow (Mullenbach et al., 2018) to preprocess the MIMIC-III dataset. 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). The only preprocessing we apply on ECtHR is to lowercase the text.

Training We fine-tune the 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:

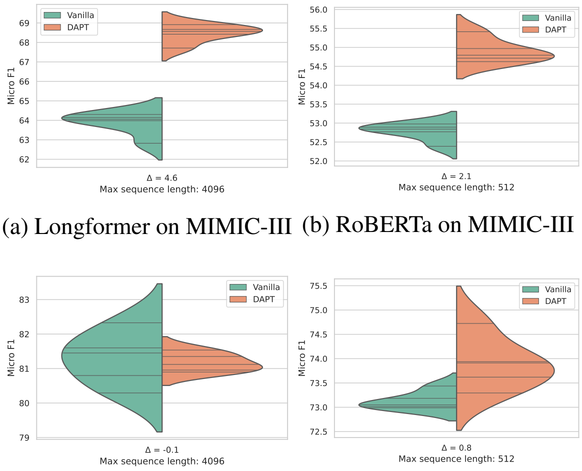
$$\mathcal{L} = \sum_{i=1}^{C} -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i),$$

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 a single Quadro RTX 6000 GPU, which has 24 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, is finally used for evaluation.

5 Experiments

We conduct a series of controlled experiments to understand the impact of design choices in Transformer-based models. Based on our empirical results, we derive practical advice of applying these models to long document classification regarding both effectiveness and efficiency. Finally, we compare our results against recently published results, demonstrating, contrary to previously-reported results, that the benefits of pre-trained Transformers also apply to long document classification.



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

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

Figure 5: Task-adaptive pre-training (right side in each plot) can improve the effectiveness (measured on the development sets) of pre-trained language models on downstream tasks. Δ: the difference between mean values of compared experiments.

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; Lee et al., 2020). However, task-adaptive pre-training (TAPT) that 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 each target dataset, 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 8.1). We find that task-adaptive pre-trained models outperform models without task-adaptive pre-training by a large margin on MIMIC-III (Figure 5 (a) and (b)), and smaller improvements are observed on ECtHR (Figure 5 (c) and (d)). We suspect this difference is because legal cases (i.e., ECtHR) have

Local window	Micro F_1	Speed	
		Train	Test
32	67.7 $\pm$ 0.3	9.8	16.1
64	68.2 $\pm$ 0.2	7.9	15.5
128	68.2 $\pm$ 0.1	6.8	13.9
256	68.3 $\pm$ 0.4	5.6	11.8
512	68.4 $\pm$ 0.4	3.3	7.8

Table 2: The impact of local attention window size in Longformer on MIMIC-III. Speed is measured using 'processed samples per second'. A similar pattern is observed on ECtHR, detailed in Appendix Table 10.

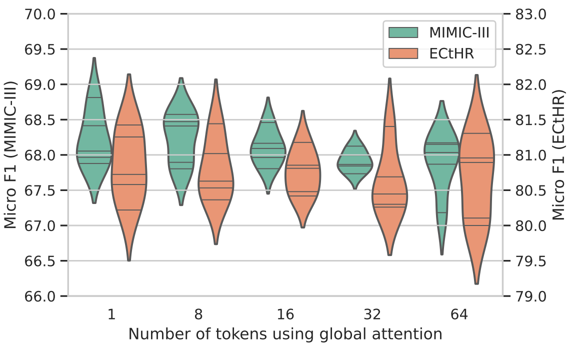


Figure 6: The effect of applying global attention on more tokens, which are evenly chosen based on their positions. In the baseline model (first column), only the [CLS] token uses global attention.

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 8.2 for a short analysis on this matter.

Take-Away #1: We suggest task-adaptive pre-training as a general first step as it is effective and cheaper than domain-adaptive pre-training. The following experiments are based on task-adaptive pre-trained Longformer and RoBERTa models.

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

Table 2 and 10 show that even using a small window size (32 tokens), the micro F_1 scores on both MIMIC-III and ECtHR development sets are still close to using a larger window size (512 tokens). A major advantage of using smaller local attention windows is the faster computation for training and evaluation. Therefore, we suggest a moderate size (64-128) of local attention window. We use a local window of 128 in the following experiments.

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 6 shows that adding more tokens using global attention does not improve performance, 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.

Regarding how to choose global tokens, the position based approach is more effective than content based (see Table 12 in the Appendix).

Take-Away #2: We suggest the following hyperparameters for Longformer for long-document classification: a local attention window of 128 tokens, and 16 equally-distributed global attention tokens.

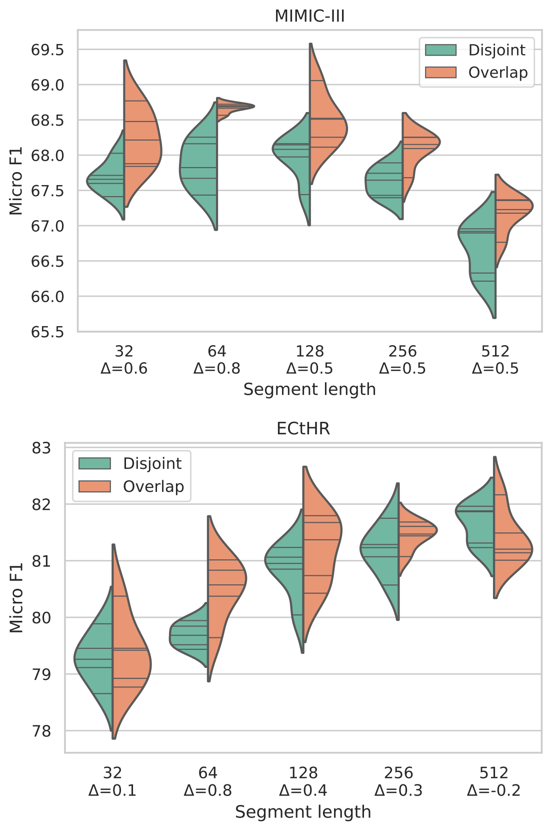


Figure 7: The effect of varying the segment length and whether allowing segments to overlap in the hierarchical Transformers. Δ: improvement due to overlap.

5.2 Hierarchical Transformers

Split documents into smaller segments. Ji et al. (2021) 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 different segment lengths and preventing context fragmentation.

Figure 7 (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.

Split documents into overlapping segments. 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 7 (right side in each violin plot) show that this simple strategy can easily improve the effectiveness of the model.

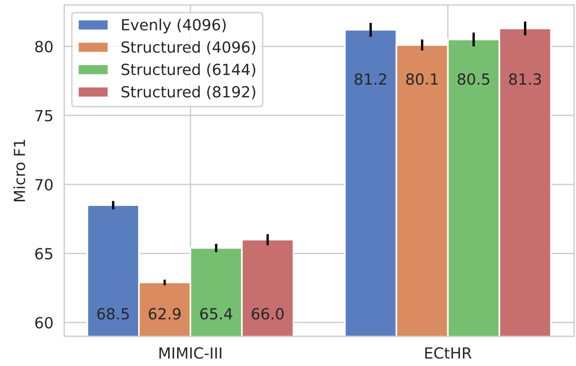


Figure 8: A comparison between evenly splitting and splitting based on document structure.

Splitting based on document structure. Chalkidis et al. (2021) 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. (2021) and use segment length (l) of 128 on ECtHR, and tune $l \in \{32, 64, 128\}$ on MIMIC-III.[6]

Figure 8 show that splitting by the paragraph-level document structure does not improve perfomance on the ECtHR dataset. On MIMIC-III, splitting based on document structure substantially underperforms evenly splitting the document.

Take-Away #3: We suggest splitting a document into small non-structure-derived segments (e.g., 128) which overlap as a starting point when employing hierarchical Transformers.

[6]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
Mullenbach et al. (2018)	$\mathbb{C}$	88.4	91.6	57.6	63.3	61.8
Dong et al. (2021)	$\mathbb{C}$	88.4	91.9	56.8	64.0	62.4
Cao et al. (2020)	$\mathbb{C}$	89.5	92.9	60.9	66.3	63.2
Li and Yu (2020)	$\mathbb{C}$	89.9	92.8	60.6	67.0	64.1
Ji et al. (2021)	$\mathbb{C}$	90.8	93.1	62.4	67.1	64.0
Xie et al. (2019)*	$\mathbb{C}$	91.4	93.6	63.8	68.4	64.4
Vu et al. (2020)*	$\mathbb{R}$	**92.5**	**94.6**	**66.6**	**71.5**	**67.5**
Transformer-based Models						
BERT (512 tokens)	$\mathbb{T}$	81.3 $\pm$ 0.3	85.0 $\pm$ 0.3	41.3 $\pm$ 1.2	52.3 $\pm$ 0.6	53.5 $\pm$ 0.2
RoBERTa (512 tokens)	$\mathbb{T}$	81.0 $\pm$ 0.2	84.8 $\pm$ 0.2	39.8 $\pm$ 0.7	52.4 $\pm$ 0.3	53.2 $\pm$ 0.2
Longformer (4096 tokens)	$\mathbb{T}$	89.9 $\pm$ 0.1	92.4 $\pm$ 0.1	60.3 $\pm$ 0.4	67.9 $\pm$ 0.3	64.8 $\pm$ 0.1
Hierarchical (4096 tokens)	$\mathbb{T}$	89.3 $\pm$ 0.2	92.0 $\pm$ 0.1	60.8 $\pm$ 0.9	67.7 $\pm$ 0.3	64.2 $\pm$ 0.3
Hierarchical (5120 tokens)	$\mathbb{T}$	89.5 $\pm$ 0.1	92.0 $\pm$ 0.1	61.7 $\pm$ 0.5	68.2 $\pm$ 0.3	64.5 $\pm$ 0.2
Transformer-based Models with Label-wise Attention Network						
Longformer (4096 tokens)	$\mathbb{T}$	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 tokens)	$\mathbb{T}$	91.1 $\pm$ 0.1	93.5 $\pm$ 0.1	63.8 $\pm$ 0.3	69.9 $\pm$ 0.2	65.3 $\pm$ 0.2
Hierarchical (5120 tokens)	$\mathbb{T}$	91.2 $\pm$ 0.1	93.6 $\pm$ 0.1	63.8 $\pm$ 0.5	70.2 $\pm$ 0.3	65.9 $\pm$ 0.2

Table 3: Comparison of state-of-the-art against our models on the MIMIC-III test set. Results are sorted by Micro F_1. $\mathbb{C}$: CNN-based models; $\mathbb{R}$: RNN-based models; and $\mathbb{T}$: Transformer-based models. Models marked with an asterisk (*) exploit the label hierarchy, i.e., they use a better classification component, as defined in Section 3.

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:

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

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

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

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

Table 15 in the Appendix shows 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.

5.4 Bringing it all together & Comparison with State of the art

We benchmark the combination of our recommendations for the Longformer and hierarchical Transformer model. Table 3 shows the results of our best-performing models against the state of the art. We find that both the Longformer and hierarchical Transformers are effective at long document classification, contrary to previous claims. Longformer, which can process up to 4096 tokens, achieves competitive results with the best performing CNN-based model (Xie et al., 2019). Note that Xie et al. and Vu et al. (2020) truncate all documents to a maximum sequence length of 4000 words ($\approx 4,932$ subtokens, see Appendix Table 5). By using label-wise attention network and processing equally long sequences, the hierarchical models outperform all CNN-based models by 1.8 points. Our Transformer-based models only underperform the RNN-based model, which additionally exploits the label hierarchy of ICD codes (Vu et al., 2020). We hypothesize that using a similar hierarchy-aware classifier could lead to comparable or even better results.

The ECtHR dataset (Chalkidis et al., 2021) is a very recently released dataset, where the authors used hierarchical Transformers. Our results are on par with their results (See Appendix Table 6).

6 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 or 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., 2021), 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.

Methods of adapting transformers for long documents can be categorised into two approaches: *recurrent* Transformers and *sparse* attention Transformers. The standard 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 tree-of-sequences LSTMs (Xie et al., 2018). Since ICD codes are organised under a hierarchical structure, methods are proposed to exploit relation between codes based on label co-occurrence (Dong et al., 2021), label count (Du et al., 2019), label hierarchical (Vu et al., 2020), knowledge graph (Xie et al., 2019; Cao et al., 2020; Lu et al., 2020), code's textual descriptions (Mullenbach et al., 2018; Xie et al., 2018; Rios and Kavuluru, 2018). More recently, Ji et al. (2021); Gao et al. (2021) investigate various methods of applying BERT on ICD coding. Different from our work, they mainly focus on comparing different domain-specific BERT models that are pre-trained on various types of corpora. Ji et al. show that PubMedBERT—pre-trained from scratch on biomedical articles—outperforms other BERT variants pre-trained on clinical notes or health-related posts; Gao et al. show that BlueBERT—pre-trained on PubMed abstracts and clinical notes—performs best. However, both report that Transformers-based models perform worse than CNN-based ones.

7 Conclusions

Transformers have previously been criticised as incapable of long-document classification. In this paper, we carefully study the role of different components of such models. By conducting experiments on MIMIC-III and ECtHR, two challenging datasets from the clinical and legal domains respectively, we draw important conclusions. Firstly, Longformer, a sparse attention model, which can process up to 4096 tokens, achieves competitive results with CNN-based models; its performance is relatively stable across different datasets; 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 different across two datasets, 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114, Online.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael J Bommarito II, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2021. LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. *arXiv*, 2110.00976.

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 *9th International Conference on Learning Representations*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.

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 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2914–2927, Online.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic.

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. *Journal of Biomedical Informatics*, 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. *Journal of the American Medical Informatics Association*, 26(11):1279–1285.

Richárd Farkas and György Szarvas. 2008. Automatic construction of rule-based ICD-9-CM coding systems. In *BMC bioinformatics*, volume 9, page S10.

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 Journal of Biomedical and Health Informatics*, 25(9).

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online.

Shaoxiong Ji, Matti Hölttä, and Pekka Marttinen. 2021. Does the Magic of BERT Apply to Medical Code Assignment? A Quantitative Study. *arXiv*, 2103.06511.

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. *Scientific Data*, 3(1):1–9.

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 *Proceedings of the 16th BioNLP Workshop*, pages 328–332, Vancouver, Canada.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*, Online.

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 informatics and decision making*, 15(1):53.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Fei Li and Hong Yu. 2020. ICD coding from clinical text using multi-filter residual convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8180–8187.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2935–2943, Online.

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

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable Prediction of Medical Codes from Clinical Text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana.

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.

Damian Pascual, Sandro Luck, and Roger Wattenhofer. 2021. Towards BERT-based Automatic ICD Coding: Limitations and Opportunities. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 54–63, Online.

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 *Biological, translational, and clinical language processing*, pages 97–104, Prague, Czech Republic.

Alan Ramponi and Barbara Plank. 2020. Neural Unsupervised Domain Adaptation in NLP—A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online).

Anthony Rios and Ramakanth Kavuluru. 2018. Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142, Brussels, Belgium. EMNLP.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification? *arXiv*, 1905.05583.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432, Lisbon, Portugal.

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 *Advances in Neural Information Processing Systems*, pages 5998–6008, Long Beach, California.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for ICD coding from clinical text. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 3335–3341, Online.

Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-Specific Document Representation for Multi-Label Text Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 466–475, Hong Kong, China.

Pengtao Xie, Haoran Shi, Ming Zhang, and Eric Xing. 2018. A Neural Architecture for Automated ICD Coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1066–1076, Melbourne, Australia.

Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. 2019. EHR Coding with Multi-scale Feature

Attention and Structured Knowledge Graph Propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 649–658.

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 *The 29th ACM International Conference on Information and Knowledge Management*, pages 1725–1734.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, San Diego, California.

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 *Advances in Neural Information Processing Systems*, pages 17283–17297.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy.

8 Appendix

8.1 Details of task-adaptive pre-training

Hyperparameters and training time for task-adaptive pre-training can be found in Table 4.

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

Table 4: Hyperparameters and training time (measured on MIMIC-III dataset) for task-adaptive pre-training Longformer and RoBERTa. Batch size = batch size per GPU $\times$ number of GPUs $\times$ gradient accumulation steps.

8.2 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 1, 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).

We suspect task-adaptive pre-training is more useful on MIMIC-III than on ECtHR (Figure 5) 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 5.

8.3 Results on ECtHR test set

Results in Table 6 show that our results are on par with the ones reported in (Chalkidis et al., 2021),

	MIMIC-III	ECtHR
Fragmentation ratio	1.233	1.118
Perplexity	1.351	1.079

Table 5: 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.

where different BERT variants are evaluated. Regarding hierarchical method, we split a document into overlapping segments, each of which has 128 tokens. We use the default setting for Longformer as in (Beltagy et al., 2020).

	Macro F_1	Micro F_1
RoBERTa	77.0	78.6
Longformer	75.8	78.8
BERT	78.3	79.6
CaseLaw-BERT	76.8	79.7
BigBird	76.9	79.9
DeBERTa	78.3	79.9
Legal-BERT	77.2	80.6
Our Models		
Hierarchical (4096 tokens)	75.5 ± 1.0	80.4 ± 0.4
Longformer (4096 tokens)	76.4 ± 1.1	80.4 ± 0.4

Table 6: Comparison of our results against the results reported in (Chalkidis et al., 2021) on the ECtHR test set. Results are sorted by Micro F_1.

8.4 A comparison between Longformer and Hierarchical model

Table 7 shows a comparison between Longformer and Hierarchical models regarding their efficiency. We set the maximum sequence length as 4096 and use 128 for both local window size in Longformer and segment length in hierarchical models. Note that we try to make full use of GPU memory (24G) via setting as large as possible batch size (i.e., training batch size of 5 and test batch size of 256 in Longformer; 7 and 256 in hierarchical model).

	Longformer	Hierarchical
# parameters	148.6M	139.0M
Training speed	6.2	12.1
Test speed	22.4	32.1

Table 7: A comparison between Longformer and Hierarchical models. Speed: processed documents per second, measured on MIMIC-III.

8.5 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.

	AUC		F_1		
Seq	Macro	Micro	Macro	Micro	P@5
MIMIC-III					
512	81.3 ± 0.3	85.2 ± 0.2	39.2 ± 1.2	52.1 ± 0.6	52.9 ± 0.4
1024	83.4 ± 0.2	87.2 ± 0.3	41.7 ± 1.1	55.6 ± 0.3	56.2 ± 0.3
2048	86.3 ± 0.3	89.6 ± 0.2	47.3 ± 1.2	60.1 ± 0.4	59.4 ± 0.5
4096	88.2 ± 0.2	91.3 ± 0.2	52.8 ± 0.8	63.9 ± 0.5	62.0 ± 0.3
ECtHR					
512	—	—	67.9 ± 2.1	73.3 ± 0.4	—
1024	—	—	72.5 ± 1.4	76.7 ± 0.5	—
2048	—	—	74.9 ± 1.7	79.3 ± 0.5	—
4096	—	—	77.6 ± 1.8	81.3 ± 0.7	—

Table 8: Detailed results of Figure 1: the effectiveness of Longformer on the MIMIC-III and ECtHR development sets.

	AUC		F_1		
	Macro	Micro	Macro	Micro	P@5
Longformer on **MIMIC-III**					
Vanilla	88.2 ± 0.2	91.3 ± 0.2	52.8 ± 0.8	63.9 ± 0.5	62.0 ± 0.3
TAPT	90.2 ± 0.2	92.6 ± 0.1	61.0 ± 0.6	68.5 ± 0.4	64.7 ± 0.2
RoBERTa on **MIMIC-III**					
Vanilla	81.4 ± 0.1	85.1 ± 0.2	39.9 ± 0.5	52.8 ± 0.2	53.4 ± 0.4
TAPT	82.5 ± 0.2	86.1 ± 0.2	43.2 ± 1.2	54.9 ± 0.3	54.7 ± 0.2
Longformer on **ECtHR**					
Vanilla	—	—	77.6 ± 1.8	81.3 ± 0.7	—
TAPT	—	—	78.1 ± 0.7	81.2 ± 0.2	—
RoBERTa on **ECtHR**					
Vanilla	—	—	67.9 ± 2.4	73.1 ± 0.2	—
TAPT	—	—	68.9 ± 1.0	73.9 ± 0.5	—

Table 9: Detailed results of Figure 5: the impact of task-adaptive pre-training. Note that we use maximum sequence length 512 for RoBERTa and 4096 for Longformer in this experiment.

	AUC		F_1		
Size	Macro	Micro	Macro	Micro	P@5
			MIMIC-III		
32	89.8 ± 0.2	92.4 ± 0.1	59.0 ± 1.0	67.7 ± 0.3	64.1 ± 0.2
64	90.0 ± 0.2	92.5 ± 0.1	60.5 ± 0.5	68.2 ± 0.2	64.5 ± 0.3
128	90.1 ± 0.1	92.5 ± 0.1	60.7 ± 0.3	68.2 ± 0.1	64.4 ± 0.2
256	90.1 ± 0.1	92.6 ± 0.1	60.6 ± 0.9	68.3 ± 0.4	64.6 ± 0.2
512	90.2 ± 0.2	92.6 ± 0.1	60.9 ± 0.8	68.4 ± 0.4	64.7 ± 0.3
			ECtHR		
32	—	—	78.3 ± 1.0	80.9 ± 0.7	—
64	—	—	77.0 ± 2.9	80.9 ± 0.3	—
128	—	—	78.5 ± 1.8	80.8 ± 0.4	—
256	—	—	78.2 ± 0.5	81.2 ± 0.3	—
512	—	—	78.1 ± 2.2	81.1 ± 0.4	—

Table 10: Detailed results of Table 2: the impact of local attention window size in Longformer.

	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 11: Detailed results of Figure 6: 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 ± 0.2	92.6 ± 0.1	60.5 ± 0.9	68.2 ± 0.3	64.7 ± 0.3
8	89.7 ± 0.2	92.0 ± 0.1	61.0 ± 1.3	66.9 ± 0.4	64.0 ± 0.4
16	89.4 ± 0.2	91.9 ± 0.1	60.1 ± 1.2	66.5 ± 0.3	63.9 ± 0.5
32	89.4 ± 0.4	91.9 ± 0.2	60.3 ± 1.6	66.4 ± 0.6	63.7 ± 0.7
64	89.1 ± 0.4	91.7 ± 0.2	59.4 ± 2.0	66.2 ± 0.7	63.4 ± 0.7
			ECtHR		
1	—	—	78.5 ± 1.8	80.8 ± 0.4	—
8	—	—	79.2 ± 0.3	80.9 ± 0.2	—
16	—	—	77.6 ± 1.2	80.4 ± 0.4	—
32	—	—	77.1 ± 0.7	80.0 ± 0.2	—
64	—	—	76.6 ± 1.1	79.9 ± 0.5	—

Table 12: 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
			Disjoint segments on **MIMIC-III**		
32	89.4 ± 0.1	92.1 ± 0.0	60.8 ± 0.5	67.7 ± 0.2	63.3 ± 0.2
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**		
32	89.7 ± 0.2	92.3 ± 0.1	61.7 ± 0.3	68.2 ± 0.4	63.7 ± 0.1
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**		
32	—	—	75.5 ± 1.7	79.3 ± 0.4	—
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**		
32	—	—	74.1 ± 2.6	79.4 ± 0.6	—
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	—

Table 13: Detailed results of Figure 7: 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 14: Detailed results of Figure 8: 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 15: The effect of label-wise attention network.