BudgetLongformer: Can we Cheaply Pretrain a SotA Legal Language Model From Scratch?

Joel Niklaus and Daniele Giofré
Thomson Reuters Labs, Zug, Switzerland
firstname.lastname@thomsonreuters.com

Abstract

Pretrained transformer models have achieved state-of-the-art results in many tasks and benchmarks recently. Many state-of-the-art Language Models (LMs), however, do not scale well above the threshold of 512 input tokens. In specialized domains though (such as legal, scientific or biomedical), models often need to process very long text (sometimes well above 10000 tokens). Even though many efficient transformers have been proposed (such as Longformer, BigBird or FNet), so far, only very few such efficient models are available for specialized domains. Additionally, since the pretraining process is extremely costly in general – but even more so as the sequence length increases – it is often only in reach of large research labs. One way of making pretraining cheaper is the Replaced Token Detection (RTD) task, by providing more signal during training, since the loss can be computed over all tokens. In this work, we train Longformer models with the efficient RTD task on legal data to showcase that pretraining efficient LMs is possible using much less compute. We evaluate the trained models on challenging summarization tasks requiring the model to summarize long texts to show to what extent the models can achieve good performance on downstream tasks. We find that both the small and base models outperform their baselines on the in-domain BillSum and out-of-domain PubMed tasks in their respective parameter range. We publish our code and models for research purposes.

1 Introduction

Pretrained transformer models have achieved excellent performance across various Natural Language Processing (NLP) tasks such as Text Classification (TC), Named Entity Recognition (NER), Question Answering (QA) and summarization (Devlin et al., 2019; Yang et al., 2020; He et al., 2021; Zhang et al., 2020a).
been shown to improve downstream performance in many domains such as law (Chalkidis et al., 2020; Xiao et al., 2021), biology (Lee et al., 2019), scientific articles (Beltagy et al., 2019), clinical documents (Li et al., 2022), or even code (Chen et al., 2021). Domain-specific pretraining coupled with the RTD task, however, has not been studied in the legal domain so far.

Depending on the domain, documents might be extremely long. Texts from the legal domain, for example, tend to span multiple pages, ranging from 10s to 100s of pages, which translates to tens of thousands tokens. The quadratic time and memory requirement of the attention typically used in the transformer architecture (Vaswani et al., 2017) prohibits efficient processing of sequences longer than 512 tokens on current hardware. A rich body of research investigates how transformers can be adapted to efficiently process longer input (Tay et al., 2020b; Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2021; Roy et al., 2021; Kitaev et al., 2020; Tay et al., 2021; Lee-Thorp et al., 2021).

Longformer (Beltagy et al., 2020) is one of these efficient transformer architectures for long sequences, leveraging windowed and global attention. So far, to the best of our knowledge, there does not yet exist a public Longformer model pretrained on English legal data\(^1\), although Xiao et al. (2021) have proven the effectiveness of the Longformer in dealing with long legal text in many Chinese-related tasks. This work aims to fill this gap.

To test the ability to grasp long-distance dependencies in the text, we mainly evaluated our Language Models (LMs) on the task of automatic (abstractive) summarization. It consists of capturing the most important concepts/ideas from the (long) document and then rewriting it in a shorter passage in a grammatical and logically coherent way (Chen et al., 2019).

In particular, we used the BillSum benchmark, as a domain-specific summarization task, obtaining a new state-of-the-art (SOTA) (see Figure 1); and the PubMed benchmark, to evaluate the model’s ability outside the legal context (i.e., in the biomedical context), obtaining comparable metrics even though the LM has only been pretrained on legal data and the tokenizer is also optimized for legal data (see Figure 2).

We emphasize that this performance was achieved with a minimal pretraining phase due to the combination of the RTD task and the Longformer infrastructure, making our LM very attractive from the point of view of building costs. For instance, our model saw only 3.2M examples during pretraining, whereas RoBERTa (Liu et al., 2019) or PEGASUS-large (Zhang et al., 2020a) saw 4.1B examples. RoBERTa was trained for 1024 GPU days, whereas our small and base models only used 12 and 24 GPU days respectively (16GB NVIDIA V100 GPUs for both models).

Since many tasks in legal NLP are formulated as TC problems, a hierarchical architecture has been used frequently to process long documents (Chalkidis et al., 2019; Niklaus et al., 2021). This simple hierarchical architecture, indeed, cannot be easily adapted to solve the more complex sequence-to-sequence tasks like token classification or summarization, because it do not take efficiently long input correlations. For this reason, in this work, we pretrain a more versatile Longformer model.

Finally, for completeness, we evaluated our LMs using the LexGLUE benchmark, which is mainly based on multi-class and multi-label legal TC problems for short texts.

### Contributions

The contributions of this paper are five-fold:

- We train and release a new model pretrained on recently published curated English legal text (Henderson et al., 2022), capable of handling input spans longer than 512 tokens out of the box.
- We apply the promising, but seldom used RTD task (Clark et al., 2020) on a Longformer model (Beltagy et al., 2020), for the first time, calling it BudgetLongformer.
- On the BillSum benchmark (Kornilova and Eidelman, 2019), our models are a new SOTA compared to models of the same size. Especially, our small model outperforms all baseline approaches, and a transformer base model (Vaswani et al., 2017) containing almost 4 times more encoder parameters (110M vs. 29M). It even outperforms the PEGASUS base model (Zhang et al., 2020a) whose encoder is also almost 4 times larger and has been pretrained specifically for the abstractive summarization task in mind.
- We verified that pretraining with the RTD task is suitable for down-stream summarization tasks by

\(^1\)On the web there is a model based on Longformer in a legal domain but no link how it was obtained and on its actual performance (https://huggingface.co/saibo/legal-longformer-base-4096).
evaluating our model on an out-of-domain benchmark (PubMed), obtaining comparable results with summarization-specific architectures.

• On the LexGLUE benchmark (Chalkidis et al., 2021), despite the obvious emphasis on covering classification tasks even for short documents, our models achieve metrics equivalent to those of architectures that are better suited to this length of document, and with substantially fewer numbers of parameters and pretraining steps.

Main Research Questions

In this work, we pose and examine five main research questions:

RQ1: Is it possible to generate an ad-hoc LM with domain (e.g. legal) expertise from scratch, reducing costs and CO2 emissions?

RQ2: Is it possible to pretrain a Longformer model with the RTD task (aka BudgetLongformer)?

RQ3: How does our BudgetLongformer compare with other models on the challenging summarization task? Particularly in the case of a legal domain-specific benchmark such as BillSum?

RQ4: How well does our BudgetLongformer generalize to other domains, for example in the biomedical domain, as evaluated by the PubMed summarization benchmark?

RQ5: How do our LMs compare with other models on the Text Classification (TC) benchmark LexGLUE?

2 Related Work

Domain-Specific Language Models

Previous work showed that domain-specific pretraining shows promising results on datasets of specialized domains such as law (Chalkidis et al., 2020; Xiao et al., 2021), biology (Lee et al., 2019), scientific articles (Beltagy et al., 2019), clinical documents (Li et al., 2022), or even code (Chen et al., 2021).

Gururangan et al. (2020) show that continued pretraining on a RoBERTa checkpoint on biomedical data, scientific articles in computer science, and reviews, clearly improves downstream performance in the respective domain-specific datasets. The effect was less pronounced on datasets from the news domain, presumably because RoBERTa has seen many news articles in its pretraining already.

Long Document Processing

In the past few years, a vast amount of research has been devoted to addressing the problem of quadratic time and memory complexity associated with the dense attention mechanism (Vaswani et al., 2017), practically limiting the maximum sequence length severely (often to 512 tokens) (Tay et al., 2020b; Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2021; Roy et al., 2021; Kitaev et al., 2020; Tay et al., 2021; Lee-Thorp et al., 2021). These research works have given rise to a new class of transformers, referred to as sparse transformers or efficient transformers (Tay et al., 2020b). Reducing the cost associated with the computation of the dense attention matrix while maintaining the same performance is the core idea behind efficient transformers. This is often achieved by introducing sparsity in the attention matrix in a variety of ways that may be fixed pattern such as local (windowed) attention (Child et al., 2019; Beltagy et al., 2020), global attention (Zaheer et al., 2021) or learnable patterns such as routing attention (Roy et al., 2021) and LSH attention (Kitaev et al., 2020) or a random pattern (Zaheer et al., 2021; Tay et al., 2021). Recently, Lee-Thorp et al. (2021) proposed to use Fourier transforms instead of the attention layer. A comprehensive list of efficient transformers and the detailed description of their attention mechanism can be found in the survey by Tay et al. (2020b). (Tay et al., 2020a) proposed a series of tasks designed for testing the capabilities of these different models suitable for longer inputs. However, this so-called “Long Range Arena” considers mostly artificial tasks, with the goal of evaluating the models independently of any pretraining.

Efficient Pretraining

ELECTRA-style pretraining (Clark et al., 2020) has been shown to reduce training cost substantially, while matching the performance of SOTA LMs. ELECTRA leverages a smaller generator model (discarded after pretraining), that changes some tokens. The larger discriminator model (used for down-stream tasks) must predict for each token if it was changed by the generator or not, similar to how Generative Adversarial Networks (GANs) are trained (Goodfellow et al., 2014). This enables the loss to be relevant for every token, leading to much faster and thus more efficient training.
3 Datasets

In this section, we briefly introduce the datasets used in our experiments.

3.1 Pile of Law

Henderson et al. (2022) recently released a large-scale English corpus suitable for pretraining LMs. It contains 256 GB of diverse legal text in English from various jurisdictions and judicial bodies including for example bills, court decisions and contracts from the US, Canada, and Europe even though the focus clearly lies on US data. While there are 28 US datasets available (253.25 GB or 99%), there is only 1 Canadian dataset (243 MB or 0.09%), 3 European datasets (2.3 GB or 0.9%) and 2 international datasets (212 MB or 0.08%). The non-US datasets only cover the categories “Legal Case Opinions and Filings”, “Laws” and “Conversations”, but do not cover categories “Legal Analyses”, “Contracts / Business Documents” and “Study Materials”, whereas the US data is much more diverse and covers all categories.

3.2 BillSum

Kornilova and Eidelman (2019) introduced a legislative summarization dataset from 21K US bills from 1993 to 2018. It is challenging due to the technical nature and complex structure of the bills. Additionally, the bills are rather long, ranging from 5K to 20K characters (∼1K to 4K tokens) with their summaries being up to 5K characters (∼1K tokens) long (see Appendix G for more details).

3.3 PubMed

Cohan et al. (2018) introduced another challenging summarization dataset in a specialized domain (scientific articles from the biomedical domain). It includes 133K scientific papers together with their abstracts in English. The papers are 3K words long on average and the summaries (abstracts) 200 words. Thus, similar to the BillSum dataset, this dataset is well suited as a test bed for methods capable of long document summarization. Note, that in this dataset the domain is vastly different from the legal domain (see Appendix G for more details).

3.4 LexGLUE

Chalkidis et al. (2021) recently introduced a benchmark for the English legal domain called LexGLUE. LexGLUE contains six TC tasks and one QA task comprising diverse legal data such as US court decisions and contracts, terms of service documents, EU legislation and cases from the European Court to Human Rights (ECtHR). There exists a public leaderboard of diverse models on GitHub, with Legal-BERT (Chalkidis et al., 2020) performing best.

The LexGLUE benchmark focuses on evaluating LMs in legal TC and QA tasks. In LexGLUE, 4 out of 7 tasks involve documents with input lengths lower than 512 tokens on average. From the remaining 3 tasks, the ECtHR A and B tasks and the SCOTUS tasks involve documents with long span, and the median of the first two is also less than 1000 tokens. Usually, legal documents are much longer than 512 tokens and thus this distribution might not be representative of real-world tasks. Shorter input length tasks may be better handled by short-input models (e.g., BERT, RoBERTa, Legal-BERT, etc.).

4 BudgetLongformer

In the legal domain, it is especially important that models can handle long input. So far, there does not exist an English legal model capable of handling more than 512 tokens. To make pretraining more affordable, we combined the well-proven Longformer model (Beltagy et al., 2020) with the RTD task proposed by Clark et al. (2020).

5 Experimental Setup

In this section, we describe how we set up the experiments. In all our experiments, we made use of AMP mixed precision training and evaluation to reduce costs and GPU memory. For all our experiments, we used the huggingface transformers library (Wolf et al., 2020) available under an Apache 2.0 license.

5.1 Tokenizer

We trained a byte-level BPE tokenizer (Wang et al., 2019) similar to Beltagy et al. (2020). To encode the complicated legal language well, we chose a relatively large vocabulary of 64K tokens (additionally, we did not apply any preprocessing/cleaning of the input texts). We trained the tokenizer using

---

2Canadian Court Opinions (ON, BC)  
3European Court of Human Rights Opinions, EUR-LEX and European Parliament Proceedings Parallel Corpus  
4World Constitutions and U.N. General Debate Corpus  
5Our experiments show that using our tokenizer one token corresponds to 5.33 characters on average.
the huggingface tokenizers library\textsuperscript{7} on the entire PileOfLaw training split (\(~192\)GB, \(~22.5\)B tokens, \(~7.5\)M documents), covering a wide array of English legal texts, mostly from the US.

### 5.2 Pretraining

We trained the caselaw models on the training subset “Court Listener Opinions” from the PileOfLaw (59.3 GB, 7.65B words, 3.39M documents). The diverse models were trained on caselaw (“Court Listener Opinions” & “Court Listener Docket Entry Documents”), legislation (“US Code”, “State Codes” & “EURLEX”) and contracts (“Atticus Contracts” & “EDGAR Contracts”). To balance the training data, we limited the number of documents to 500K (this affects Court Listener Opinions, Court Listener Docket Entry Documents and EDGAR Contracts. Please see Table 1 for more details. Our validation set consisted of 1000 randomly selected examples from the respective training set.\textsuperscript{8}

To maximally use the available data, we concatenated all the examples and then cut them off in slices of the model’s maximum sequence length (4096). We did this in batches of 1000 examples with multiprocessing to speed up data preparation. The last slice in each batch will not contain 4096 tokens, so we dropped it.

We trained both a small (29M parameters) and a base (159M parameters) model for each configuration. To reach 100K steps it took a bit less than 3 days for the small model and a bit less than 6 days for the base model on 4 16GB NVIDIA V100 GPUs. The achieved training and evaluation losses are shown in Table 7 in Appendix C. Interestingly, we find that the diverse models achieve lower train-

### 5.3 Downstream Benchmarks

#### BillSum

When finetuning on the BillSum dataset (Kornilova and Eidelman, 2019) we trained using early stopping with patience of 3 epochs. We paired our pretrained encoder model with a randomly initialized bart-base decoder model (Lewis et al., 2020).\textsuperscript{9} We used a batch size of 32 and learning rate of 7e-5 after tuning in {5e-4, 9e-5, 7e-5, 5e-5, 3e-5, 1e-5}. We used the bart-base default config for num_beams (4) and no_repeat_ngram_size (3). We set the maximum input length to 1024 and the maximum target length to 256 to save compute. However, many summaries get cut off at 256 tokens. This is why we took our best model and trained it with maximum input length 4096 and maximum target length 1024 (see results in Table 4 and examples in Table 10). Due to high training costs, we only trained it with one random seed (42). Our models contain 29M (small) and 159M (base) parameters in the encoder and 96M parameters in the decoder resulting in a total of 125M (small) and 255M (base) parameters.

#### PubMed

Additionally, we evaluated on the PubMed summarization task (Cohan et al., 2018) using the same settings as for the BillSum task. We set the maximum input length to 4096 and the maximum generation length to 512.

\textsuperscript{7}https://github.com/huggingface/tokenizers
\textsuperscript{8}We used such a relatively small validation set to save compute.

\textsuperscript{9}Interestingly, the randomly initialized decoder yielded better results than when we used the weights from the pre-trained huggingface checkpoint at https://huggingface.co/facebook/bart-base.
LexGLUE

Finally, we evaluated on LexGLUE (Chalkidis et al., 2021) using the publicly available scripts without modification to ensure consistent and comparable results. Because of compute limitations, we ran each experiment with only one random seed (1) and with the default set of hyperparameters. We speculate that hyperparameter tuning could further improve the performance of the proposed model.

6 Results

In the following three sections, we present the results on the BillSum dataset, the PubMed dataset and the LexGLUE benchmark. Tables 2 and 3 in Appendix A compare the models evaluated on the summarization and LexGLUE benchmarks, respectively.

6.1 BillSum

Our results on the BillSum dataset are presented in Figure 1 and Table 4 in Appendix B.

We observe that even our small diverse model clearly exceeds the baseline of the original article (DOC + SUM), even though their model is based on BERT-large which contains almost 12 times more encoder parameters and has been pretrained for 10 times more steps. Even more surprisingly, our small diverse model is on par with the PEGASUS-base model (Zhang et al., 2020a) (37.58 vs. 37.78 Rouge-L), pretrained using the Gap-Sentences task specifically designed for abstractive summarization. Furthermore, their model contains almost 4 times more encoder parameters and has seen 40 times more training examples during pretraining (128M vs. 3.2M; see Table 2 in Appendix A).

By scaling up our model to the base size, we even approach the performance of PEGASUS-large (40.5 vs. 45.8 Rouge-L). PEGASUS-large has seen three orders of magnitude more training examples during its pretraining in comparison to our model (4.1B vs. 3.2M) and contains more than twice as many encoder parameters (340M vs. 159M).

We conclude that pretraining with the RTD task is highly effective, with minimal compute for long-input summarization in-domain.

6.2 PubMed

Our results on the PubMed dataset are presented in Figure 2 and Table 5 in Appendix B.

Similar to the results on BillSum, our small model clearly outperforms the Transformer-base model (23.24 vs. 19.02 Rouge-L) and approaches the PEGASUS-base model (23.24 vs. 25.2 Rouge-L) even though we did not specifically pretrain our model for summarization and our model has seen 40 times fewer examples during pretraining (3.2M vs. 128M). Similar again, we almost reach the performance of PEGASUS-large (26.53 vs. 27.69 Rouge-L) while having seen 1280 times fewer examples during pretraining (3.2M vs. 4.1B).

Note, that we pretrain on a much narrower domain than PEGASUS (legal text vs. C4). Our tokenizer and model has never seen medical data during its pretraining phase. Finally, our tokenizer has 1/3 fewer tokens than the PEGASUS tokenizer (64K vs. 96K).

In conclusion, pretraining with the RTD task is even effective on an out-of-domain downstream summarization task.

6.3 LexGLUE

Table 3 in Appendix A compares the models evaluated on the LexGLUE benchmark. Note, that these models differ strongly on many dimensions such as the number and types of training steps, the
Our results on the LexGLUE benchmark are presented in Table 6 in Appendix B and in Figures 3 and 4 for the small and base models respectively. Figure 5 in Appendix B shows all the models evaluated on LexGLUE combined.

From the results shown in Table 6, we can observe that our models do not improve on the SOTA for short input length tasks. This suggests that for such tasks a more accurate description of the first 512 tokens, obtained through a pretraining dataset with a comparable distribution of token inputs, is more appropriate. This could be an explanation for why our base model is not able to beat the trained models in the short input length.

Despite the previous statement, we can also note that there is quite a clear correlation between the Micro-F1 and the number of parameters of the model in the case of small-size models. LegalBERT-small is an exception, outperforming DistilBERT but having fewer parameters. But LegalBERT-small has been pretrained on the same data as is contained in 6 out of 7 LexGLUE tasks. It is also likely, that the test sets have been contained in the pretraining data. Our small model is still in this trend of performance to model size, despite having seen much fewer examples during pretraining (almost 200 times fewer than BERT-Tiny). While in the case of the base model, this trend is still true for the same samples seen, if we leave out Legal-BERT and CaseLaw-BERT for the reasons already expressed. This suggests that potentially extending the pretraining dataset with also short documents might improve the performance of our model in this regime as well. In our case, we avoided focusing too much on this point since the purpose of the paper is to solve the legal long documents as input.

Finally, we did not tune the hyperparameters at all. It is well known that proper hyperparameter tuning and already selecting the right random seeds can significantly influence the downstream performance (Liu and Wang, 2021; Dodge et al., 2020). Note that especially our small models, like BERT-Tiny and miniLM, lag behind in the UnfairToS task (Macro-F1 score below 15). This could be due to an unlucky random seed (Mosbach et al. (2021) and Dodge et al. (2020) reported training performance strongly dependent on the random seed).

7 Conclusions and Future Work

7.1 Answers to Main Research Questions

RQ1: Is it possible to generate an ad-hoc LM with domain (e.g., legal) expertise from scratch, reducing costs and CO₂ emissions? Yes, we showcase in this work that it is possible to pretrain a domain-expertise LM from scratch with minimal compute, achieving comparable performance with methods that have seen more than three orders of magnitude more pretraining examples. Especially when there is no well-performing large teacher model available, our method is advisable.

RQ2: Is it possible to pretrain a Longformer model with the RTD task (aka BudgetLongformer)? Yes, in this work, we show that it is possible to pretrain a Longformer model with the RTD task.

RQ3: How does our BudgetLongformer compare with other models on the challenging summarization task? Particularly in the case of a legal domain-specific benchmark such as BillSum? Our LMs compare favorably to baselines on the challenging domain-specific summarization benchmark BillSum, requiring the models to process long inputs. Our small model outperforms the larger PEGASUS-base model, and our base model almost reaches the performance of the larger PEGASUS-large model. Both baselines have been pretrained with much more compute and data, and additionally with a pretraining task crafted specifically for summarization.

RQ4: How well does our BudgetLongformer generalize to other domains, for example in the biomedical domain, as evaluated by the PubMed summarization benchmark? Yes, our results on
the out-of-domain PubMed summarization benchmark show that our models compare favorably to baselines. Again, our small model outperforms PEGASUS-base and our base model approaches the performance of PEGASUS large.

**RQ5:** How do our LMs compare with other models on the understanding classification benchmark LexGLUE? Our small models compare favorably to baselines in their respective parameter range. Our base models approach the performance of the baselines even though (a) we trained using significantly less compute, (b) we did not pretrain on short documents, and (c) we did not tune the hyperparameters at all.

### 7.2 Limitations

ELECTRA-style training has the disadvantage of the setup being slightly more complicated, requiring a generator and a discriminator. Additionally, the generator should be smaller than the discriminator to ensure stable training. This makes it difficult to warm start from available checkpoints, since two models of different sizes are required. Often, small models are not released, which makes it difficult to warm-start base models using the RTD task. We leave the direction of warm starting a large discriminator with a base generator to future work.

Except for EUR-LEX (1.31 GB or 1.8% of our diverse dataset), our models have only seen US data during the pretraining phase. So, while these models are expected to work well on US data or datasets with similar content such as heavily influenced by the US or mainly common-law based, legal data from Europe for example is expected to look very different (mainly civil-law based except for the UK) and often translated from the original European languages. Thus, our models are not expected to transfer well to such kind of data.

Because of insufficient compute, we were not able to scale up our models in terms of parameter size, batch size and number of pretraining steps. So while we can show that our approach scales well from the small to the base model, it is unknown if this continues to even larger model sizes. Although it is expected to produce better results, we do not know if using a higher batch size and more pretraining steps boosts performance significantly. Additionally, the lacking compute budget made evaluating on more and especially large datasets like BigPatent impossible. Therefore, we cannot give any conclusions at this point to whether our results are robust across a wide range of datasets.

So far, we did not evaluate our summarization models using newer metrics such as BERTScore (Zhang et al., 2020b) or BARTScore (Yuan et al., 2021). However, our baselines only evaluated using ROUGE, so we would have needed to rerun the baseline experiments to be able to compare our results to on these newer scores.

So far, we did not have the resources to conduct a thorough human expert evaluation of the quality of our summarization outputs. Such an evaluation would be needed for production systems and for better comparison of models. However, it also requires highly educated medical experts (for PubMed) or lawyers with specific expertise in US bills (for BillSum) respectively, and thus a prohibitively high amount of resources.

For comparing the efficiency of pretraining, the number of FLOPs would probably be best. We compared the models’ efficiency based on the number of seen examples during pretraining, due to ready availability (most papers report the batch size and the number of steps, but few papers report the FLOPs). Liu et al. (2019) for example, also report the number of GPU days used which we can also compare to. Devlin et al. (2019), however, trained using TPUs, which makes the comparison difficult again.

### 7.3 Conclusion

In this work, we show that we can successfully pretrain Longformer models with the RTD task. Using very little pretraining we can achieve SOTA performance on the challenging legal summarization task BillSum, outperforming PEGASUS, that has been pretrained specifically for summarization. Our model even outperforms PEGASUS on the out-of-domain PubMed dataset involving biomedical research articles. To sum up, we present a simple and extremely cheap way of pretraining a long-context LM in cases without the availability of a large teacher model.

### 7.4 Future Work

Future work could test these models on further legal downstream tasks such as CUAD (Hendrycks et al., 2021) or the recently released MultiLexSum (Shen et al., 2022). Additionally, one can test whether the out-of-domain results hold on other out-of-domain summarization datasets, such as BigPatent (Sharma et al., 2019) or ArXiv (Cohan et al., 2018).
Future work could further scale up the models in terms of batch size, number of pretraining steps, number of parameters and amount of data to test what further gains can be achieved.

Due to compute constraints, we were unable to train the models long enough to reach SOTA performance on LexGLUE. Future work could take our approach further and investigate the performance to be gained by investing more compute.

Additionally, to save even more compute and to produce better models, one could investigate how to warm-start an ELECTRA pretraining from existing checkpoints. The difficulty, of course, lies in getting a suitable generator and discriminator trained with the same tokenizer. One possible setup might be Longformer-base as the generator and Longformer-large as the discriminator.

Finally, one can investigate the use of other efficient transformers with the RTD task.

Ethics Statement

Pretraining language models is a very compute-heavy process and thus leaves a large carbon footprint (Strubell et al., 2019; Patterson et al., 2021). Our method makes significantly reduces the compute requirements and thus the carbon footprint.

As with any large LM there is the risk of it producing biased or unfair output. Researchers using the model should put into place respective safeguards to identify biased and/or toxic language.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. arXiv:1903.10676 [cs]. ArXiv: 1903.10676.

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

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural Legal Judgment Prediction in English. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. arXiv:2010.02559 [cs]. ArXiv: 2010.02559.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael James Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2021. LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. SSRN Scholarly Paper ID 3936759, Social Science Research Network, Rochester, NY.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374 [cs]. ArXiv: 2107.03374.

Yangbin Chen, Yun Ma, Xudong Mao, and Qing Li. 2019. Multi-Task Learning for Abstractive and Extractive Summarization. Data Science and Engineering, 4(1):14–23.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating Long Sequences with Sparse Transformers. arXiv:1904.10509 [cs, stat]. ArXiv: 1904.10509.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. arXiv:2003.10555 [cs]. ArXiv: 2003.10555.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs]. ArXiv: 1810.04805.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. ArXiv:2002.06305 [cs].
Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. ArXiv:1406.2661 [cs, stat].

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. arXiv:2004.10964 [cs]. ArXiv: 2004.10964.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. arXiv:2111.09543 [cs]. ArXiv: 2111.09543.

Peter Henderson, Mark S. Krass, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. 2022. Pile of Law: Learning Responsible Data Filtering from the Law and a 256GB Open-Source Legal Dataset. ArXiv:2207.00220 [cs].

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. ArXiv:2103.06268 [cs].

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Kamal raj Kanakarajan, Bhuvana Kundumani, and Malaikannan Sankarasubbu. 2021. BioELECTRA:Pretrained Biomedical text Encoder using Discriminators. In Proceedings of the 20th Workshop on Biomedical Language Processing, pages 143–154, Online. Association for Computational Linguistics.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. arXiv:2001.04451 [cs, stat]. ArXiv: 2001.04451.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A Corpus for Automatic Summarization of US Legislation. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

Jinyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, page btx682. ArXiv: 1901.08746.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2021. FNet: Mixing Tokens with Fourier Transforms. arXiv:2105.03824 [cs]. ArXiv: 2105.03824.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online. Association for Computational Linguistics.

Yikuan Li, Ramsey M. Wehbe, Faraz S. Ahmad, Hanyin Wang, and Yuan Luo. 2022. Clinical-Longformer and Clinical-BigBird: Transformers for long clinical sequences. arXiv:2201.11838 [cs]. ArXiv: 2201.11838.

Xueqing Liu and Chi Wang. 2021. An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-trained Language Models. 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 2286–2300, Online. Association for Computational Linguistics.

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 [cs]. ArXiv: 1907.11692.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines. arXiv:2006.04884 [cs, stat]. ArXiv: 2006.04884.

Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. 2021. Swiss-Judgment-Prediction: A Multilingual Legal Judgment Prediction Benchmark. In Proceedings of the Natural Legal Language Processing Workshop 2021, pages 19–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon Emissions and Large Neural Network Training. arXiv:2104.10350 [cs]. ArXiv: 2104.10350.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. page 12.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient Content-Based Sparse Attention with Routing Transformers. Transactions of the Association for Computational Linguistics, 9:53–68. Place: Cambridge, MA Publisher: MIT Press.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs]. ArXiv: 1910.01108.
A Overview of Compared Models

In this section, we show detailed overviews of the model specifics (Tables 2 and 3).
Table 2: Abbreviations: P.: Pretraining, BS: Batch Size, Enc.: Encoder, Params: Parameters. Comparison of the models evaluated on the summarization tasks BillSum and PubMed.

| Model Name                  | Source                     | P. Steps (K) | P. BS | D. Steps (K) | D. BS | WS Steps (K) | WS BS | # P. Examples (M) | # Enc. Params (M) | Max Seq Len | Vocab Size (K) | PubMed Rouge-L | BillSum Rouge-L |
|-----------------------------|----------------------------|--------------|-------|--------------|-------|--------------|-------|-------------------|------------------|-------------|----------------|----------------|----------------|
| DOC + SUM                   | (Kornilova and Eidelman, 2019) | 1000         | 256   | 256          | 540   | 512          | 30    | 33.73             | Transformer-base  | 1000        | 256           | 110 1024       | 19.02          | 30.08          |
| PE-DXLARX-base               | (Zhang et al., 2020a)      | 500          | 256   | 128          | 110   | 1024         | 96    | 25.23             | PE-DXLARX-base    | 500         | 256           | 110 1024       | 27.69          | 45.8           |
| PE-DXLARX-large-C4           | (Zhang et al., 2020a)      | 500          | 8192  | 4096         | 340   | 1024         | 96    | 25.23             | BudgetLongformer  | 500         | 32            | 3.2            | 29             | 4096          | 64             | 23.24          | 37.88          |
| BudgetLongformer base diverse| ours                      | 100          | 52    | 3.2          | 29    | 4096         | 64    | 23.24             | BudgetLongformer  | 500         | 32            | 3.2            | 159            | 4096          | 64             | 26.53          | 40.50          |

Table 3: Abbreviations: P.: Pretraining, D.: Distillation, WS: Warm Start, BS: Batch Size, Params: Parameters. Comparison of the models evaluated on LexGLUE. In cases where we were not able to find the batch size in the papers, we assumed it to be 256, since this is the most widely used batch size in pretraining and the default for BERT. For DistilBERT we were not able to find the number of distillation steps, so we assumed 500K steps.

| Model Name                  | Source                     | P. Steps (K) | P. BS | D. Steps (K) | D. BS | WS Steps (K) | WS BS | # P. Examples (M) | # Param (M) | Max Seq Len | Vocab Size (K) | LexGLUE Micro-F1 |
|-----------------------------|----------------------------|--------------|-------|--------------|-------|--------------|-------|-------------------|-------------|-------------|----------------|----------------|
| small models                |                            |              |       |              |       |              |       |                   |             |             |                |                |
| BERT                        | (Devlin et al., 2019)      | 1000         | 296   | 1400         | 256   | 614.4        | 4.4   | 512               | 110         | 30          | 31             | 78.1           |
| ReBERT                      | (Lee et al., 2019)         | 1000         | 296   | 400          | 256   | 350.4        | 3.1   | 512               | 110         | 30          | 50             | 74.8           |
| DistilBERT                  | (Sanh et al., 2019)        | 1000         | 296   | 256          | 256   | 364          | 3.1   | 512               | 110         | 30          | 26.53          | 75.2           |
| Longformer small caselaw    | (Sailakis et al., 2020)    | 1000         | 296   | 256          | 256   | 256          | 3.2   | 29                | 29           | 4096        | 64             | 73.9           |
| Longformer small diverse    | ours                       | 100          | 32    | 3.2          | 29    | 4096         | 64    | 73.9              | BudgetLongformer  | 100         | 32            | 3.2            | 159            | 4096          | 64             | 73.4           |
| base models                 |                            |              |       |              |       |              |       |                   |             |             |                |                |
| BERT                        | (Devlin et al., 2019)      | 1000         | 296   |              |       | 256          | 110   | 512               | 30          | 77.8        |                |                |
| RoBERTa                     | (Liu et al., 2019)         | 1000         | 256   |              |       | 4096         | 125   | 512               | 30          | 77.8        |                |                |
| DeBERT                      | (He et al., 2021)          | 1000         | 256   |              |       | 256          | 139   | 512               | 126         | 76.3        |                |                |
| BigBird                     | (Zhou et al., 2021)        | 500          | 8192  |              |       | 398          | 256   | 4224              | 177         | 76.2        |                |                |
| Longformer                  | (Balogu et al., 2020)      | 500          | 8192  |              |       | 65           | 64    | 4010.16           | 149         | 76.5        |                |                |
| Legal-BERT base             | (Sailakis et al., 2020)    | 1000         | 296   |              |       | 256          | 110   | 512               | 31          | 79.8        |                |                |
| CasLaw-BERT                 | (Zhang et al., 2021)       | 1000         | 256   |              |       | 512          | 130   | 512               | 30          | 79.4        |                |                |
| BudgetLongformer base caselaw| ours                      | 100          | 32    | 3.2          | 195   | 4096         | 64    | 76.0              | BudgetLongformer  | 100         | 32            | 3.2            | 159            | 4096          | 64             | 76.0           |
| BudgetLongformer base diverse| ours                      | 100          | 32    | 3.2          | 195   | 4096         | 64    | 76.0              | BudgetLongformer  | 100         | 32            | 3.2            | 159            | 4096          | 64             | 76.0           |

Figure 5: Results on the LexGLUE benchmark (all models). Note that the x-axis is in log-scale.

B Detailed Results
In this section, we show detailed and comprehensive results of the compared models (Tables 4, 5 and 6 and Figure 5).

C Pretraining Details
In this section, we show additional details regarding the pretraining process (Table 7).

D Hyperparameters and Training Details
In this section, we present additional details regarding the chosen hyperparameters.

D.1 Pretraining
We pretrained our models with batch size 32 and learning rate 5e-4 and 3e-4 for the small and base models respectively. We used a Longformer attention window of 256. As described in by Clark et al. (2020), we used 10000 warm up steps and a 4 and 3 times smaller generator than the discriminator in the small and base version respectively. In contrast to Clark et al. (2020) we reduced the generator’s depth (number of hidden layers) instead of its width (embedding size, hidden size and intermediate size). We used a MLM probability of 25% for the generators.

For running the pretraining, we used an AWS p3.8xlarge instance with 4 16GB NVIDIA V100 GPUs. Training the four models to 100K steps each, took approx. 18 days or 72 GPU days in total. Previous debug runs additionally consumed approx. 3 days or 12 GPU days.

D.2 Downstream Benchmarks
Overall, we found the diverse models to be more robust in finetuning with less failed runs and typically higher performance.

For running the finetuning experiments, we used an AWS p3.16xlarge instance with 8 16GB NVIDIA V100 GPUs. Running the BillSum, PubMed, and LexGLUE experiments including hyperparameter tuning took approximately 25, 7, and 11 GPU days in total respectively.
Table 4: Results on the BillSum dataset. Enc. Params is short for Encoder Parameters.

| Model (max-in-len->max-gen-len) | # Enc. Params ↓ | Rouge-1 ↑ | Rouge-2 ↑ | Rouge-L ↑ |
|-------------------------------|----------------|-----------|-----------|-----------|
| DOC + SUM (BERT large)         | 340M           | 40.80     | 23.83     | 33.73     |
| Transformer base               | 110M           | 44.05     | 21.30     | 30.98     |
| PEGASUS base                   | 110M           | 51.42     | 29.68     | 37.78     |
| PEGASUS large (C4)             | 468M           | 57.20     | 39.56     | 45.80     |
| PEGASUS large (HugeNews)       | 468M           | 57.31     | 40.19     | 45.82     |
| BudgetLongformer small diverse (1024->128) | 29M | 53.61 | 33.54 | 42.50 |
| BudgetLongformer small diverse (1024->256) | 29M | 49.85 | 29.63 | 37.58 |
| BudgetLongformer base diverse (1024->256) | 159M | 52.70 | 32.97 | 40.50 |
| BudgetLongformer base diverse (1024->128) | 159M | 54.87 | 35.63 | 44.21 |
| BudgetLongformer base diverse (4096->1024) | 159M | 55.45 | 36.68 | 43.23 |

Table 5: Results on the PubMed dataset. Enc. Params is short for Encoder Parameters.

| Model (max-in-len->max-gen-len) | # Enc. Params ↓ | Rouge-1 ↑ | Rouge-2 ↑ | Rouge-L ↑ |
|-------------------------------|----------------|-----------|-----------|-----------|
| Transformer base               | 110M           | 33.94     | 7.43      | 19.02     |
| PEGASUS base                   | 110M           | 39.98     | 15.15     | 25.23     |
| PEGASUS large (C4)             | 468M           | 45.49     | 19.90     | 27.69     |
| PEGASUS large (HugeNews)       | 468M           | 45.09     | 19.56     | 27.42     |
| BudgetLongformer small diverse (4096->512) | 29M | 34.98 | 13.56 | 23.24 |
| BudgetLongformer base diverse (4096->512) | 159M | 41.16 | 18.15 | 26.53 |

E Library Versions

We used the following versions to the libraries in a pip requirements.txt format:
- datasets==2.4.0
- huggingface-hub==0.9.0
- nltk==3.7
- pandas==1.3.5
- rouge-score==0.1.2
- scikit-learn==1.0.2
- scipy==1.7.3
- tokenizers==0.12.1
- torch==1.12.1
- tqdm==4.64.0
- transformers==4.21.1

F Examples

Example summaries are displayed in Tables 8, 9, 10, 11, and 12. Since the documents are very long sometimes, we truncated them to the first 2500 characters. We sorted the examples by RougeL scores and show the bottom 5%, bottom 25%, top 75% and top 95% percentile.
| Model          | Data       | # Steps | Train Loss | Eval Loss |
|---------------|------------|---------|------------|-----------|
| small caselaw | 50K        | 14.61   | 15.78      |
| small caselaw | 100K       | 13.93   | 15.07      |
| small diverse | 50K        | 13.75   | 12.70      |
| small diverse | 100K       | 12.78   | 11.66      |
| base caselaw  | 50K        | 12.40   | 13.76      |
| base caselaw  | 100K       | 11.67   | 12.99      |
| base diverse  | 50K        | 10.70   | 10.01      |
| base diverse  | 100K       | 9.86    | 9.22       |

Table 7: Training and Evaluation losses for the different trained models. Note that these losses are the addition of the loss of the generator and the loss of the discriminator. Since the loss of the discriminator is much smaller, it is scaled by a factor of 50 to stabilize training.
in 35-year-old white woman of scots and english descent developed reddish urine for several days followed by eruption of vesicles and blisters on the dorsal surfaces of her hands and fingers, the sides of her nose, and her upper anterior chest, knees, and legs. She reported a longstanding hypertension, andromoia, and frequent exposure to skin dermatitis agents. A family history was noncontributory, and she denied any history of photodynamic disease. Her exposure to ultraviolet radiation was documented and found to be minimal. A skin biopsy specimen was taken from a well-defined dermal and epidermal lesion located on the right side of her hand and revealed a marked neutrophilic infiltrate. No significant photodynamic disease was observed. A skin biopsy specimen from a well-defined dermal and epidermal lesion located on the right side of her hand and revealed a marked neutrophilic infiltrate. No significant photodynamic disease was observed.

We report the first evidence of antibodies to west nile virus (wnv) in horses and humans in cuba. these findings provide evidence that wnv and slev may co-circulate in cuba.

Objectives This study aimed to determine the effect of ultrasound guidance on the accuracy of needle placement, clinical outcomes, and cost-effectiveness in comparison with conventional needle guidance. Our primary outcome was the accuracy of needle placement. Our secondary outcomes were the pain experienced during the procedure, the patient’s satisfaction with the procedure, and the cost-effectiveness of the two methods.

Materials and Methods We conducted a randomized controlled trial comparing conventional needle guidance with ultrasound guidance for needle placement in patients with symptomatic neck masses. We randomized 100 patients into two groups: the ultrasound guidance group (n=50) and the conventional needle guidance group (n=50). The ultrasound group received ultrasound guidance during the entire procedure, while the control group received conventional needle guidance.

Results The ultrasound group had a statistically significant higher accuracy of needle placement compared to the control group (p<0.05). The pain experienced during the procedure was lower in the ultrasound group compared to the control group (p<0.05). The patient’s satisfaction with the procedure was higher in the ultrasound group compared to the control group (p<0.05). There was no significant difference in the cost-effectiveness between the two groups.

Conclusion Ultrasound guidance improves the accuracy of needle placement, reduces patient pain, and increases patient satisfaction compared to conventional needle guidance. Ultrasound guidance is a cost-effective method for needle placement in patients with symptomatic neck masses.

Table 12: Examples of the PubMed dataset using the model pubmed-4096-512 base diverse
G Data Details

We used our own tokenizer to calculate the number of tokens. In Tables 6, and 7 we show the data length distributions for the BillSum train and test splits. In Tables 8, 9, and 10 we show the data length distributions for the PubMed train, validation and test splits.
Figure 6: Histograms for the BillSum training set (18949 samples).

Figure 7: Histograms for the BillSum test set (3269 samples).
Figure 8: Histograms for the PubMed train set (119924 samples).

(a) Input Text
Mean: 3044, Median: 2572
75-Quant: 3996, 95-Quant: 7057, Max: 109759

(b) Summary
Mean: 202, Median: 208
75-Quant: 262, 95-Quant: 326, Max: 391

Figure 9: Histograms for the PubMed validation set (6633 samples).

(a) Input Text
Mean: 3112, Median: 2609
75-Quant: 4011, 95-Quant: 6968, Max: 119269

(b) Summary
Mean: 203, Median: 209
75-Quant: 263, 95-Quant: 330, Max: 518

Figure 10: Histograms for the PubMed test set (6658 samples).

(a) Input Text
Mean: 3093, Median: 2596
75-Quant: 3964, 95-Quant: 6985, Max: 48750

(b) Summary
Mean: 205, Median: 213
75-Quant: 265, 95-Quant: 329, Max: 506