LEGAL-BERT: The Muppets straight out of Law School
Supplementary material
Ilias Chalkidis †‡
Prodromos Malakasiotis †‡
Nikolaos Aletras ∗
Ion Androutsopoulos †‡
† Department of Informatics, Athens University of Economics and Business
‡ Institute of Informatics & Telecommunications, NCSR “Demokritos”
∗ Computer Science Department, University of Sheffield, UK
[ihalk,fergadiotis,rulller,ion]@aueb.gr
n.aletras@sheffield.ac.uk

A Legal NLP datasets
Bellow are the details of the legal NLP datasets we used for the evaluation of our models:

- **EURLEX57K** (Chalkidis et al., 2019b) contains 57k legislative documents from EURLEX with an average length of 727 words. All documents have been annotated by the Publications Office of EU with concepts from EUROVOC.\(^1\) The average number of labels per document is approx. 5, while many of them are rare. The dataset is split into training (45k), development (6k), and test (6k) documents.

- **ECHR-CASES** (Chalkidis et al., 2019a) contains approx. 11.5k cases from ECHR’s public database. For each case, the dataset provides a list of facts. Each case is also mapped to articles of the Human Rights Convention that were violated (if any). The dataset can be used for binary classification, where the task is to identify if there was a violation or not, and for multi-label classification where the task is to identify the violated articles.

- **CONTRACTS-NER** (Chalkidis et al., 2017, 2019d) contains approx. 2k US contracts from EDGAR. Each contract has been annotated with multiple contract elements such as title, parties, dates of interest, governing law, jurisdiction, amounts and locations, which have been organized in three groups (contract header, dispute resolution, lease details) based on their position in contracts.

B Implementation details and results on downstream tasks
Below we describe the implementation details for fine-tuning BERT and LEGAL-BERT on the three downstream tasks:

**EURLEX57K**: We replicate the experiments of Chalkidis et al. (2019c), where a linear layer with L (number of labels) sigmoid activations was placed on top of BERT’s [CLS] final representation. We follow the same configuration for all LEGAL-BERT variations.

**ECHR-CASES**: We replicate the best method of Chalkidis et al. (2019a), which is a hierarchical version of BERT, where initially a shared BERT encodes each case fact independently and produces N fact embeddings ([CLS] representations). A self-attention mechanism, similar to Yang et al. (2016), produces the final document representation. A linear layer with softmax activation gives the final scores.

**CONTRACTS-NER** We replicate the experiments of Chalkidis et al. (2019d) in all of their three parts (contract header, dispute resolution, lease details). In these experiments, the final representations of the original BERT for all (sentencepiece) tokens in the sequence are fed to a linear CRF layer.

We again follow Chalkidis et al. (2019c,a,d) in the reported evaluation measures.

C Efficiency comparison for various BERT-based models
Recently there has been a debate on the over-parameterization of BERT (Kitaev et al., 2020; Rogers et al., 2020). Towards that directions most studies suggest a parameter sharing technique (Lan et al., 2019) or distillation of BERT by decreasing the number of layers (Sanh et al., 2019). However the main bottleneck of transformers in modern hardware is not primarily the total number of parameters, misinterpreted into the number of stacked layers. Instead Out Of Memory (OOM) issues mainly happen as a product of wider models.
in terms of hidden units’ dimensionality and the number of attention heads, which affects gradient accumulation in feed-forward and multi-head attention layers (see Table 1). Table 1 shows that LEGAL-BERT-SMALL despite having $3 \times$ and $2 \times$ the parameters of ALBERT and ALBERT-LARGE has faster training and inference times. We expect models overcoming such limitations to be widely adopted by researchers and practitioners with limited resources. Towards the same direction Google released several lightweight versions of BERT.

<table>
<thead>
<tr>
<th>Model</th>
<th>Params</th>
<th>$T$</th>
<th>$HU$</th>
<th>$AH$</th>
<th>Max BS</th>
<th>Training Speed</th>
<th>Inference Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-BASE</td>
<td>110M</td>
<td>12</td>
<td>768</td>
<td>12</td>
<td>6</td>
<td>1.00x</td>
<td>1.00x</td>
</tr>
<tr>
<td>ALBERT</td>
<td>125M</td>
<td>12</td>
<td>768</td>
<td>12</td>
<td>12</td>
<td>1.20x</td>
<td>1.21x</td>
</tr>
<tr>
<td>ALBERT-LARGE</td>
<td>180M</td>
<td>24</td>
<td>1024</td>
<td>12</td>
<td>4</td>
<td>0.95x</td>
<td>0.97x</td>
</tr>
<tr>
<td>DISTIL-BERT</td>
<td>66M</td>
<td>6</td>
<td>768</td>
<td>12</td>
<td>16</td>
<td>1.66x</td>
<td>2.30x</td>
</tr>
<tr>
<td>LEGAL-BERT</td>
<td>130M</td>
<td>12</td>
<td>768</td>
<td>12</td>
<td>6</td>
<td>1.00x</td>
<td>1.00x</td>
</tr>
<tr>
<td>LEGAL-BERT-SMALL</td>
<td>35M</td>
<td>6</td>
<td>512</td>
<td>8</td>
<td>26</td>
<td>2.43x</td>
<td>4.00x</td>
</tr>
</tbody>
</table>

Table 1: Comparison of BERT-based models for different batch sizes ($BS$) in a single 11GB NVIDIA-2080Ti. Resource efficiency of the models mostly relies on the number of hidden units ($HU$), attentions heads ($AH$) and Transformer blocks $T$, rather than the number of parameters.

References


2https://github.com/google-research/bert