Abstract

Prior research notes that BERT’s computational cost grows quadratically with sequence length thus leading to longer training times, higher GPU memory constraints and carbon emissions. While recent work seeks to address these scalability issues at pre-training, these issues are also prominent in fine-tuning especially for long sequence tasks like document classification. Our work thus focuses on optimizing the computational cost of fine-tuning for document classification. We achieve this by complementary learning of both topic and language models in a unified framework, named TopicBERT. This significantly reduces the number of self-attention operations – a main performance bottleneck. Consequently, our model achieves a 1.4x (~40%) speedup with ~40% reduction in CO₂ emission while retaining 99.9% performance over 5 datasets.

1 Introduction

Natural Language Processing (NLP) has recently witnessed a series of breakthroughs by the evolution of large-scale language models (LM) such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019) etc. due to improved capabilities for language understanding (Bengio et al., 2003; Mikolov et al., 2013). However this massive increase in model size comes at the expense of very high computational costs: longer training time, high GPU/TPU memory constraints, adversely high carbon footprints, and unaffordable invoices for small-scale enterprises.

Figure 1 shows the computational cost (training time: millisecond/batch; CO₂ emission, and GPU memory usage) of BERT all of which grow quadratically with sequence length (N). We note that this is primarily due to self-attention operations. Moreover, as we note in Table 1, the staggering energy cost is not limited to only the pre-training stage but is also encountered in the fine-tuning stage when processing long sequences as is needed in the task of document classification. Note that the computational cost incurred can be quite significant especially because fine-tuning is more frequent than pre-training and BERT is increasingly used for processing long sequences. Therefore, this work focuses on reducing computational cost in the fine-tuning stage of BERT especially for the task of document classification.

Recent studies address the excessive computational cost of large language models (LMs) in the pre-training stage using two main compression techniques: (a) Pruning (Michel et al., 2019; Lan et al., 2020) by reducing model complexity, and (b) Knowledge Distillation (Hinton et al., 2015; Tang et al., 2019; Turc et al., 2019; Sanh et al., 2019a) which a student model (compact model) is trained
to reproduce a teacher (large model) leveraging the teacher’s knowledge. Finally, in order to process long sequences, Xie et al. (2019) and Joshi et al. (2019) investigate simple approaches of truncating or partitioning them into smaller sequences, e.g., to fit within 512 token limit of BERT for classification; However, such partitioning leads to a loss of discriminative cross-partition information and is still computationally inefficient. In our work, we address this limitation by learning a complementary representation of text using topic models (TM) (Blei et al., 2003; Miao et al., 2016; Gupta et al., 2019). Because topic models are bag-of-words based models, they are more computationally efficient than large scale language models that are contextual. Our work thus leverages this computational efficiency of TMs for efficient and scalable fine-tuning for BERT in document classification.

Specifically our contributions:

1. **Complementary Fine-tuning**: We present a novel framework: *TopicBERT*, i.e., topic-aware BERT that leverages the advantages of both neural network-based TM and Transformer-based BERT to achieve improved document-level understanding. We report gains in document classification task with full self-attention mechanism and topical information.

2. **Efficient Fine-tuning**: *TopicBERT* offers an efficient fine-tuning of BERT for long sequences by reducing the number of self-attention operations and jointly learning with TM. We achieve a 1.4x (~40%) speedup while retaining 99.9% of classification performance over 5 datasets. Our approaches are model agnostic; therefore we extend BERT and DistilBERT models. Code is available at https://github.com/YatinChaudhary/TopicBERT.

**Carbon footprint (CO$_2$) estimation**: We follow Lacoste et al. (2019) and use ML CO$_2$ Impact calculator$^1$ to estimate the carbon footprint (CO$_2$) of our experiments using the following equation:

$$CO_2 = \text{Power consumption} \times \text{Time (in hours)} \times \text{Carbon produced by local power grid}$$

where, Power consumption = 0.07KW for NVIDIA Tesla T4 16 GB Processor and Carbon produced by local power grid = 0.61 kg CO$_2$/kWh. Therefore, the final equation becomes:

$$CO_2 = 0.07kW \times \text{Time (in hours)} \times 0.61 \times 1000 \text{ gram eq. CO}_2/\text{kWh} \quad (1)$$

$^1$https://mlco2.github.io/impact/

---

**Figure 2**: Topic-aware BERT (*TopicBERT*): Joint fine-tuning of NVDM and BERT; The input in BERT is $D$ for complementary fine-tuning while $D_t$ ($t^{th}$ partition of $D$) for complementary+efficient fine-tuning. $+$: addition; $\circ$: Hadamard product; $\oplus$: concatenation; Green dashed lines: variational component of NVDM.

In Figure 1, we run BERT for different sequence lengths (32, 64, 128, 256 and 512) with batch-size=4 to estimate GPU-memory consumed and CO$_2$ using equation 1. We run each model for 15 epochs and compute run-time (in hours).

For Table 1, we estimate CO$_2$ for document classification tasks (BERT fine-tuning) considering 512 sequence length. We first estimate the total BERT fine-tuning time in terms of research activities and/or its applications beyond using multiple factors. Then, using equation 1 the final CO$_2$ is computed. (See supplementary for detailed computation)

### 2 Methodology: TopicBERT

Figure 2 illustrates the architecture of *TopicBERT* consisting of: (1) Neural Topic Model (NTM), (2) Neural Language Model (NLM) to achieve complementary and efficient document understanding.

#### 2.1 TopicBERT: Complementary Fine-tuning

Given a document $D = [w_1, ..., w_N]$ of sequence length $N$, consider $V \in \mathbb{R}^Z$ be its BoW representation, $v_i \in \mathbb{R}^Z$ be the one-hot representation of the word at position $i$ and $Z$ be the vocabulary size.

The **Neural Topic Model** component (Figure 2, left) is based on Neural Variational Document Model (NVDM) (Miao et al., 2016), seen as a variational autoencoder for document modeling in an unsupervised generative fashion such that:
(a) an MLP encoder \( f^{MLP} \) and two linear projections \( l_1 \) and \( l_2 \) compress the input document \( V \) into a continuous hidden vector \( h_{TM} \in \mathbb{R}^K \):

\[
\begin{align*}
\pi &= g(f^{MLP}(V)) \text{ and } e \sim \mathcal{N}(0, I) \\
\mu(V) &= l_1(\pi) \text{ and } \sigma(V) = l_2(\pi) \\
q(h_{TM}|V) &= \mathcal{N}(h_{TM}; \mu(V), \text{diag}(\sigma(V))) \\
h_{TM} \sim q(h_{TM}|V) \implies h_{TM} = \mu(V) \oplus \epsilon \odot \sigma(V)
\end{align*}
\]

The \( h_{TM} \) is sampled from a posterior distribution \( q(h_{TM}|V) \) that is parameterized by mean \( \mu(V) \) and variance \( \sigma(V) \), generated by neural network. We call \( h_{TM} \) as a document-topic-representation (DTR), summarizing document semantics.

(b) a softmax decoder \( \hat{V} \), i.e., \( p(V|h_{TM}) = \prod_{i=1}^{N} p(v_i|h_{TM}) \) reconstructs the input document \( V \) by generating all words \( \{v_i\} \) independently:

\[
\begin{align*}
p(v_i|h_{TM}) &= \frac{\exp[h_{TM}^T U_{i} + c_i]}{\sum_{j=1}^{2} \exp[h_{TM}^T U_{j} + c_j]} \\
L_{NVDM} &= \mathbb{E}_{q(h_{TM}|V)}[\log p(V|h_{TM})] \text{ - KLD}
\end{align*}
\]

where \( U \in \mathbb{R}^{K \times Z} \) and \( c \in \mathbb{R} \) are decoding parameters, \( L_{NVDM} \) is the lower bound, i.e., \( \log p(V) \geq L_{NVDM} \) and \( \text{KLD} = \text{KL}[q(h_{TM}|V) || p(h_{TM})] \) is the KL-Divergence between the Gaussian posterior \( q(h_{TM}|V) \) and prior \( p(h_{TM}) \) for \( h_{TM} \). During training, NVDM maximizes log-likelihood \( \log p(V) = \sum_{h_{TM}} p(V|h_{TM})p(h_{TM}) \) by maximizing \( L_{NVDM} \) using stochastic gradient descent. See further details in Miao et al. (2016).

The Neural Language Model component (Figure 2, right) is based on BERT (Devlin et al., 2019). For a document \( D \) of length \( N \), BERT first tokenizes the input sequence into a list of sub-word tokens \( X \) and then performs \( O(N^2 n_t) \) self-attention operations in \( n_t \) encoding layers to compute its contextualized representation \( o_{CLS} \in \mathbb{R}^{H_B} \), extracted via a special token [CLS]. Here, \( H_B \) is the number of hidden units. We use \( o_{CLS} \) to fine-tune BERT.

Complementary Learning: TopicBERT (Figure 2) jointly performs neural topic and language modeling in a unified framework, where document-topic \( h_{TM} \) and contextualized \( o_{CLS} \) representations are first concatenated-projected to obtain a topic-aware contextualized representation \( h_p \in \mathbb{R}^{H_B} \) and then \( h_p \) is fed into a classifier:

\[
\begin{align*}
h_p &= (h_{TM} \oplus o_{CLS}) \odot P \\
p(y = y_i|D) &= \frac{\exp[h_p^T Q_{Y} y_i + b_y]}{\sum_{y_j} \exp[h_p^T Q_{Y} y_j + b_y]} \\
L_{TopicBERT} &= \alpha \log p(y = y_i|D) + (1 - \alpha) L_{NVDM}
\end{align*}
\]

where, \( P \in \mathbb{R}^{H_B \times H_B} \) is the projection matrix, \( H = H_B, Q \in \mathbb{R}^{H_B \times L} \) & \( b \in \mathbb{R}^{L} \) are classification parameters, \( y_i \in \{y_1, ..., y_L\} \) is the true label for \( D \) and \( L \) is the total number of labels. During training, the TopicBERT maximizes the joint objective \( L_{TopicBERT} \) with \( \alpha \in (0, 1) \). Similarly, we extract \( o_{CLS} \) from DistillBERT (Sanh et al., 2019a) and the variant is named as TopicDistilBERT.

2.2 TopicBERT: Efficient Fine-tuning

Since the computation cost of BERT grows quadratically \( O(N^2) \) with sequence length \( N \) and is limited to 512 tokens, therefore there is a need to deal with larger sequences. The TopicBERT model offers efficient fine-tuning by reducing the number of self-attention operations in the BERT component.

In doing this, we split a document \( D \) into \( p \) partitions each denoted by \( D' \) of length \( N/p \). The NVDM component extracts document-topic representation \( h_{TM} \) efficiently for the input \( D \) and BERT extracts contextualized representation \( o_{CLS} \) for \( D' \), such that the self-attention operations are reduced by a factor of \( p^2 \) in each batch while still modeling all cross-partition dependencies within the complementary learning paradigm. Table 2 illustrates the computation complexity of BERT vs TopicBERT and the efficiency achieved.

### 3 Experimental Results and Analysis

**Datasets:** For document classification, we use 5 datasets (Reuter18, Imdb, 20NS, Ohsumed, AGnews) from several domains. (See supplementary for data descriptions and experimental results of AGnews)

**Baselines:** (a) CNN (Kim, 2014), (b) BERT-Avg: Logistic classifier over the vector \( D_B \) of a document obtained by averaging its contextualized word embeddings from BERT, (c) BERT-Avg+DTR: Logistic classifier over concatenation(\( D_B, DTR \)) where \( DTR = h_{TM} \) from pre-trained NVDM, i.e., no joint fine-tuning, (d) DistillBERT (Sanh et al., 2019b), (e) BERT fine-tuned. We compare our ex-

| Sequence length | BERT | TopicBERT |
|-----------------|------|-----------|
| Time Complexity (batch-wise) | \( b(N^2 H_B/n_t) \) | \( bKZ + b(N^2 H_B/p^2 n_t) \) |
| #Batches | \( n_b \) | \( n_b \times p \) |
| Time Complexity (epoch-wise) | \( b(N^2 H_B n_t) \) | \( bKZ n_b + b(N^2 H_B n_t/p) n_l \) |

Table 2: Time complexity of BERT vs TopicBERT. Here, \( b \): batch-size, \( n_b \): #batches and \( n_t \): #layers in BERT. Note, the compute cost of NVDM and self-attention operations as \( KZ \ll (N^2 H_B/p) n_t \). In TopicBERT: \( p = 1 \) for complementary learning, and \( p = \{2, 4, 8\} \) for complementary-efficient learning.
**Table 3:** TopicBERT for document classification (macro-F1). Retn: Retention in F1 vs BERT; $T_{epoch}$: average epoch time (in hours); $T$: $T_{epoch} \times 15$ epochs; CO2: Carbon in gram eq. (equation 1); bold: Best (fine-tuned BERT-variant) in column; underlined: Most efficient TopicBERT-x vs BERT; Gain (performance): TopicBERT-x vs BERT; Gain (efficiency): underlined vs BERT

**Experimental setup:** For BERT component, we split the input sequence $D$ into $p$ equal partitions each of length $x = N_B/p$, where $N_B = 512$ (due to token limit of BERT) and $p \in \{1, 2, 4, 8\}$ (a hyperparameter of TopicBERT). To avoid padding in the last partition, we take the last $x$ tokens of $D$. We run TopicBERT-x (i.e., BERT component) for different sequence length ($x$) settings, where (a) $p = 1$, i.e., TopicBERT-512 denotes complementary fine-tuning, and (b) $p \in \{2, 4, 8\}$, i.e., TopicBERT-\{256, 128, 64\} denotes complementary-efficient fine-tuning. Note, NVDIM always considers the full-sequence. We execute 3 runs of each experiment on an NVIDIA Tesla T4 16 GB Processor to a maximum of 15 epochs. Carbon footprint (CO2) is computed as per equation 1. (See supplementary for hyperparameters)

**Results:** Table 3 illustrates gains in performance and efficiency of TopicBERT, respectively due to complementary and efficient fine-tuning. E.g. in Reuters8, TopicBERT-512 achieves a gain of 1.6% in F1 over BERT and also outperforms DistilBERT.

In the efficient setup, TopicBERT-128 achieves a significant speedup of 1.9× (1.9× reduction in CO2) in fine-tuning while retaining (Retn) 99.25% of F1 of BERT. For IMDB and 20NS, TopicBERT-256 reports similar performance to BERT, however with a speedup of 1.2× and also outperforms DistilBERT in F1 though consuming similar time $T_{epoch}$: Additionally, TopicBERT-512 exceeds DistilBERT in F1 for all the datasets. At $p = 8$, TopicBERT-64 does not achieve expected efficiency perhaps due to saturated GPU-parallelization (a trade-off in decreasing sequence length and increasing #batches).

Overall, TopicBERT-x gains in: (a) performance: 1.604%, 0.850%, 0.537%, 0.260% and 0.319% in F1 for Reuters8, 20NS, IMDB, Ohsumed and AGnews (in supplementary), respectively, and (b) efficiency: a speedup of 1.4× (~40%) and thus, a reduction of ~40% in CO2 over 5 datasets while retaining 99.9% of F1 compared to BERT. It suggests that the topical semantics improves document classification in TopicBERT (and TopicDistilBERT: a further 1.55x speedup in Distil-
**Analysis (Interpretability):** For two different input documents, Figure 3 illustrates the misclassification by BERT and correct classification by TopicBERT explained by the top key terms of dominant topic in DTR.

**Analysis (Pareto Frontier):** As shown in Table 3, gains in TopicBERT has been analyzed on two different fronts: (a) gain on the basis of performance (F1 score), and (b) gain on the basis of efficiency (Fine-tuning time/\(CO_2\)). Figure 4 illustrates the following Pareto frontier analysis plots for Reuters8 dataset: (a) \(F1\) score vs Fine-tuning time (left), and (b) \(F1\) score vs \(CO_2\) (carbon footprint) (right). Here green dashed line represents Pareto frontier connecting optimal solutions.

We have presented two novel architectures: TopicBERT and TopicDistilBERT for an improved and efficient (Fine-tuning time/\(CO_2\)) document classification, leveraging complementary learning of topic (NVDM) and language (BERT) models.

**Acknowledgments**

This research was supported by Bundeswirtschaftsministerium (bmi.de), grant 01MD19003E (PLASS (plass.io)) at Siemens AG - CT Machine Intelligence, Munich Germany.
A Supplementary Material

A.1 CO$_2$: Carbon footprint estimation

For Table 1, we estimate CO$_2$ for document classification tasks (BERT fine-tuning) considering 512 sequence length. We first estimate the frequency of BERT fine-tuning in terms of research activities and/or its application beyond. We estimate the following items:

1. Number of scientific papers based on BERT = 5532 (number of BERT citations to date: 01, June 2020)
2. Conference acceptance rate: 25% (i.e., 4 times the original number of submissions or research/application beyond the submissions)
3. Average number of datasets used = 5
4. Average run-time of 15 epochs in fine-tuning BERT over 5000 documents (Reuters8-sized data) of maximum 512 sequence length = 12 hours on the hardware-type used

Therefore, using equation 1 in main paper,

\[
\text{CO}_2 \text{ estimate in fine-tuning } BERT = 0.07 \times (5532 \times 4 \times 5) \times 12 \times 0.61 \text{ kg eq.} = 56,692 \times 2,0462 \text{ lbs eq} = 124,985 \text{ lbs eq.}
\]

A.2 Data statistics and preprocessing

Table 4 shows data statistics of 5 datasets used for 4. Average run-time of 15 epochs in fine-tuning BERT over 5000 documents (Reuters8-sized data) of maximum 512 sequence length = 12 hours on the hardware-type used

Therefore, using equation 1 in main paper,

\[
\text{CO}_2 \text{ estimate in fine-tuning } BERT = 0.07 \times (5532 \times 4 \times 5) \times 12 \times 0.61 \text{ kg eq.} = 56,692 \times 2,0462 \text{ lbs eq} = 124,985 \text{ lbs eq.}
\]

A.3 Experimental setup

Table 5 and 7 shows hyperparameter settings of NVDM and BERT components of our proposed TopicBERT model for document classification task. We initialize BERT component with pretrained BERT-base model released by Devlin et al. (2019).

Fine-tuning of TopicBERT is performed as follows: (1) perform pretraining of NVDM component, (2) initialize BERT component with BERT-base model, (3) perform complementary + efficient fine-tuning for 15 epochs, using joint loss objective:

\[
\mathcal{L}_{\text{TopicBERT}} = \alpha \log p(y = y_i|D) + (1 - \alpha)\mathcal{L}_{\text{NVDM}}
\]

where, \(\alpha \in \{0.1, 0.5, 0.9\}\). For CNN, we follow the experimental setup of Kim (2014).

A.4 Results of TopicBERT for AGnews

Table 8 shows gains in performance and efficiency of TopicBERT vs BERT for AGnews dataset. TopicBERT achieves: (a) a gain of 0.3% in F1 (perfor-
Table 6: TopicDistilBERT vs DistilBERT for document classification (macro-F1) in complementary (TopicDistilBERT-{512}) and efficient (TopicDistilBERT-{256, 128}) learning setup. Here, Rtn: Retention in F1 vs BERT; $T_{epoch}$: average epoch time (in hours); $T$: $T_{epoch}$×15 epochs; CO2: Carbon footprint in gram eq. (equation 1); bold: Best (fine-tuned DistilBERT-variant) in column; underlined: Most efficient TopicDistilBERT-x vs DistilBERT; Gain (performance): TopicDistilBERT-x vs DistilBERT; Gain (efficiency): underlined vs DistilBERT

| Models | Reuters8 (news domain) | 20NS (news domain) |
|--------|------------------------|---------------------|
|        | F1 | Rtn | $T_{epoch}$ | T | CO2 | F1 | Rtn | $T_{epoch}$ | T | CO2 |
| CNN    | 0.852 ± 0.000 91.12% | 0.007 0.340 14.51 | 0.786 ± 0.000 95.50% | 0.109 1.751 74.76 |
| DistilBERT | 0.934 ± 0.003 100.00% | 0.129 1.938 82.75 | 0.816 ± 0.005 100.00% | 0.313 1.700 200.69 |
| TopicDistilBERT-512 | 0.941 ± 0.007 100.75% | 0.132 1.976 84.47 | 0.820 ± 0.000 100.49% | 0.320 4.810 205.38 |
| TopicDistilBERT-256 | 0.943 ± 0.006 100.96% | 0.085 1.272 54.31 | 0.802 ± 0.000 98.28% | 0.190 2.850 121.69 |
| TopicDistilBERT-128 | 0.911 ± 0.012 97.57% | 0.096 1.444 61.66 | 0.797 ± 0.000 97.67% | 0.387 5.800 247.66 |
| Gain (performance) | - | 0.964 % | - | 0.490% | - | - | - | - | - |
| Gain (efficiency) | - | 100.96% | - | 1.5× | - | 98.28% | - | 1.6× | - | 1.6× |

Table 7: Hyperparameters search and optimal settings for BERT component of TopicBERT used in the experimental setup for document classification. $\dag$ → additional hyperparameter introduced for joint modeling in TopicBERT; $\dagger$ → N = 32 is only used for AGnews dataset; (*) → hyperparameter values taken from pretrained BERT-base model released by Devlin et al. (2019).

Table 8: TopicBERT for document classification (macro-F1) for AGnews dataset. Rtn: Retention in F1 vs BERT; $T_{epoch}$: average epoch time (in hours); $T$: $T_{epoch}$×15 epochs; CO2: Carbon footprint in gram eq. (equation 1); bold: Best (fine-tuned BERT-variant) in column; underlined: Most efficient TopicBERT-x vs BERT; Gain (performance): TopicBERT-x vs BERT; Gain (efficiency): underlined vs BERT

A.5 TopicDistilBERT vs DistilBERT

Table 6 reports scores of TopicDistilBERT vs DistilBERT for two datasets (Reuters8 and 20NS). We follow the similar schemes of sequence lengths (512, 256 and 128) to evaluate the performance of the (a) complementary learning via TopicDistilBERT-512 vs DistilBERT, and (b) efficient learning via TopicDistilBERT-{256, 128} vs DistilBERT.

For Reuters8 in complementary setup, TopicDistilBERT-512 achieves a gain (0.941 vs 0.934) in F1 over DistilBERT. In the efficient setup, TopicDistilBERT-256 achieves a significant speedup of 1.3× (1.5×, i.e., ~50% reduction in CO2) in fine-tuning while retaining (Rtn) 100.96% of F1 of DistilBERT.

For 20NS in complementary setup, TopicDistilBERT-512 achieves a gain (0.820 vs 0.816) in F1 over DistilBERT. In the efficient setup, TopicDistilBERT-256 achieves a speedup of 1.6× (1.6×, i.e., ~60% reduction in CO2).
Nobel honours sub-atomic world
US scientists David Gross, David Politzer and Frank Wilczek win the Nobel physics prize for their insights into the deep structure of matter.

Figure 5: Interpretability analysis of document classification for AGnews dataset (for 2 input documents): Illustration of document misclassification by BERT model and correct classification by TopicBERT explained by the top key terms of dominant topic in DTR.

Additionally, TopicBERT-512 exceeds DistilBERT in F1 for the two datasets. At p = 4, TopicDistilBERT-128 does not achieve expected efficiency perhaps due to saturated GPU-parallelization (a trade-off in decreasing sequence length and increasing #batches) and therefore, we do not partition further.

Overall, TopicDistilBERT-x achieves gains in: (a) performance: 0.964%, and 0.490% in F1 for Reuters8 and 20NS, respectively, and (b) efficiency: a speedup of $1.55 \times$ ($\sim 55\%$) and thus, a reduction of $\sim 55\%$ in CO$_2$ over 2 datasets while retaining 99.6% of F1 compared to DistilBERT baseline model.

It suggests that the topical semantics improves document classification in TopicDistilBERT (and TopicBERT) and its energy-efficient variants. Based on our two extensions: TopicBERT and TopicDistilBERT, we assert that our proposed approaches of complementary learning (fine-tuning) are model agnostic of BERT models.

A.6 Interpretability Analysis in TopicBERT

To analyze the gain in performance (F1 score) of TopicBERT vs BERT, Figure 5 shows document label misclassifications due to BERT model. However, TopicBERT model is able to correctly predict the labels using document topic representation (DTR) which explains the correct predictions by the top key terms of the dominant topic discovered.