TopicBERT for Energy Efficient Document Classification

***Yatin Chaudhary[1,2], *Pankaj Gupta[1],**
Khushbu Saxena[3], Vivek Kulkarni[4], Thomas Runkler[3], Hinrich Schütze[2]
[1]DRIMCo GmbH, Munich, Germany | [2]CIS, University of Munich (LMU), Munich, Germany
[3]Siemens AG, Munich, Germany | [4]Stanford University, California, USA
yatin.chaudhary@drimco.net

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 ($\sim$ 40%) speedup with $\sim$ 40% reduction in CO_2 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), XL-Net (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; CO_2 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.

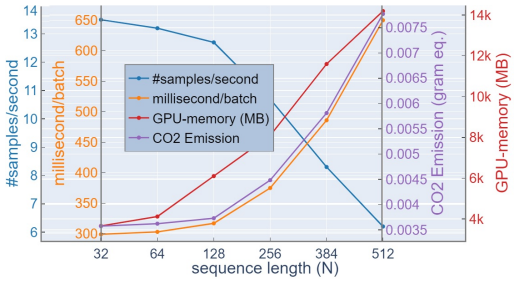

Figure 1: Computational cost vs sequence length

	CO_2
BERT pre-training (NAS) (Strubell et al., 2019)	626k
BERT fine-training (n=512)*	+ 125k

Table 1: Similar to Strubell et al. (2019) who estimate the carbon footprint of BERT during pretraining, we estimate the carbon footprint (lbs of CO_2 equivalent) during finetuning BERT for document classification. *: see *supplementary* material for details.

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

*Equal Contribution

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1682–1690

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 an 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 ($\sim$ 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 in 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/

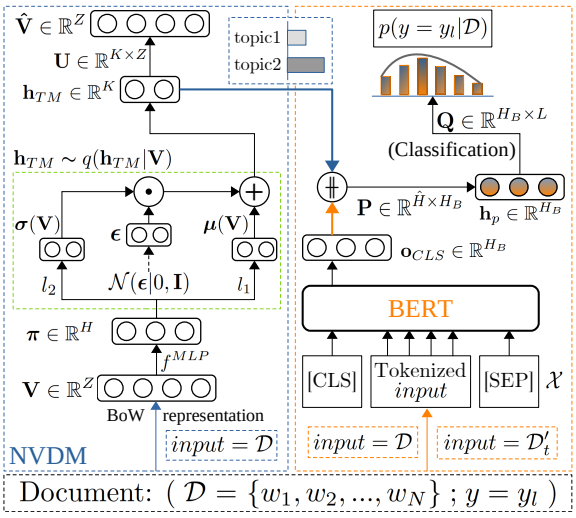

Figure 2: Topic-aware BERT (*TopicBERT*): Joint fine-tuning of NVDM and BERT; The *input* in BERT is $\mathcal{D}$ for complementary fine-tuning while $\mathcal{D}'_t$ (t^{th} partition of $\mathcal{D}$) for complementary+efficient fine-tuning. $\oplus$: addition; $\odot$: Hadamard product; $\textcircled{\scriptsize ||}$: 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 $\mathcal{D} = [w_1, ..., w_N]$ of sequence length N, consider $\mathbf{V} \in \mathbb{R}^Z$ be its BoW representation, $\mathbf{v_i} \in \mathbb{R}^Z$ 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 $\mathbf{V}$ into a continuous hidden vector $\mathbf{h}_{TM} \in \mathbb{R}^K$:

$$\boldsymbol{\pi} = g(f^{MLP}(\mathbf{V})) \quad \text{and} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$$
$$\boldsymbol{\mu}(\mathbf{V}) = l_1(\boldsymbol{\pi}) \quad \text{and} \quad \boldsymbol{\sigma}(\mathbf{V}) = l_2(\boldsymbol{\pi})$$
$$q(\mathbf{h}_{TM}|\mathbf{V}) = \mathcal{N}(\mathbf{h}_{TM}|\boldsymbol{\mu}(\mathbf{V}), \text{diag}(\boldsymbol{\sigma}(\mathbf{V})))$$
$$\mathbf{h}_{TM} \sim q(\mathbf{h}_{TM}|\mathbf{V}) \implies \mathbf{h}_{TM} = \boldsymbol{\mu}(\mathbf{V}) \oplus \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}(\mathbf{V})$$

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

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

$$p(\mathbf{v_i}|\mathbf{h}_{TM}) = \frac{\exp\{\mathbf{h}_{TM}^T \mathbf{U}_{:,i} + \mathbf{c_i}\}}{\sum_{j=1}^{Z} \exp\{\mathbf{h}_{TM}^T \mathbf{U}_{:,i} + \mathbf{c_i}\}}$$
$$\mathcal{L}_{NVDM} = \mathbb{E}_{q(\mathbf{h}_{TM}|\mathbf{V})}\big[\log p(\mathbf{V}|\mathbf{h}_{TM})\big] - \text{KLD}$$

where $\mathbf{U} \in \mathbb{R}^{K \times Z}$ and $\mathbf{c} \in \mathbb{R}^Z$ are decoding parameters, $\mathcal{L}_{NVDM}$ is the lower bound, i.e., $\log p(\mathbf{V}) \geqslant \mathcal{L}_{NVDM}$ and KLD = KL$[q(\mathbf{h}_{TM}|\mathbf{V})||p(\mathbf{h}_{TM})]$ is the KL-Divergence between the Gaussian posterior $q(\mathbf{h}_{TM}|\mathbf{V})$ and prior $p(\mathbf{h}_{TM})$ for $\mathbf{h}_{TM}$. During training, NVDM maximizes log-likelihood $\log p(\mathbf{V}) = \sum_{\mathbf{h}_{TM}} p(\mathbf{V}|\mathbf{h}_{TM})p(\mathbf{h}_{TM})$ by maximizing $\mathcal{L}_{NVDM}$ 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 $\mathcal{D}$ of length N, BERT first tokenizes the input sequence into a list of sub-word tokens $\mathcal{X}$ and then performs $O(N^2 n_l)$ self-attention operations in n_l encoding layers to compute its contextualized representation $\mathbf{o}_{CLS} \in \mathbb{R}^{H_B}$, extracted via a special token [CLS]. Here, H_B is the number of hidden units. We use $\mathbf{o}_{CLS}$ to fine-tune BERT.

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

$$\mathbf{h}_p = (\mathbf{h}_{TM} \oplus \mathbf{o}_{CLS}) \cdot \mathbf{P}$$
$$p(y = y_l|\mathcal{D}) = \frac{\exp\{\mathbf{h}_p^T \mathbf{Q}_{:,y} + \mathbf{b}_y\}}{\sum_{j=1}^{L} \exp\{\mathbf{h}_p^T \mathbf{Q}_{:,y_j} + \mathbf{b}_{y_j}\}}$$
$$\mathcal{L}_{TopicBERT} = \alpha \log p(y = y_l|\mathcal{D}) + (1 - \alpha)\mathcal{L}_{NVDM}$$

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

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

Table 2: Time complexity of BERT vs TopicBERT. Here, b: batch-size, n_b: #batches and n_l: #layers in BERT. Note, the compute cost of NVDM and self-attention operations as $KZ << (N^2 H_B/p)n_l$. In TopicBERT: $p = 1$ for complementary learning, and $p = \{2, 4, 8\}$ for complmenrary+efficient learning.

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 $\mathcal{D}$ into p partitions each denoted by $\mathcal{D}'$ of length N/p. The NVDM component extracts document-topic representation $\mathbf{h}_{TM}$ efficiently for the input $\mathcal{D}$ and BERT extracts contextualized representation $\mathbf{o}_{CLS}$ for $\mathcal{D}'$, 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 (*Reuter8*, *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 $\mathcal{D}_B$ of a document obtained by averaging its contextualized word embeddings from *BERT*, (c) *BERT-Avg+DTR*: Logistic classifier over concatenation($\mathcal{D}_B$, *DTR*) where $DTR = \mathbf{h}_{TM}$ from pre-trained NVDM, i.e., no joint fine-tuning, (d) *DistilBERT* (Sanh et al., 2019b), (e) *BERT* fine-tuned. We compare our ex-

	Models	**Reuters8** (news domain)					**Imdb** (sentiment domain)				
		F1	*Rtn*	T_{epoch}	T	CO_2	*F1*	*Rtn*	T_{epoch}	T	CO_2
baselines	*CNN*	0.852 ± 0.000	91.123%	0.007	0.340	14.51	0.884 ± 0.000	94.952%	0.201	2.010	85.83
	BERT-Avg	0.882 ± 0.000	94.331%	-	0.010	0.47	0.883 ± 0.000	94.844%	-	0.077	3.29
	BERT-Avg + DTR	0.867 ± 0.000	92.727%	-	0.015	0.68	0.894 ± 0.000	96.026%	-	0.114	4.87
	DistilBERT	0.934 ± 0.003	99.893%	0.129	1.938	82.75	0.910 ± 0.003	97.744%	**0.700**	**10.500**	**448.35**
	BERT	0.935 ± 0.012	100.00%	0.208	3.123	133.34	0.931 ± 0.002	100.00%	0.984	14.755	630.04
proposal	*TopicBERT-512*	**0.950** ± 0.005	**101.60**%	0.212	3.183	135.93	0.934 ± 0.002	100.32%	1.017	15.251	651.22
	TopicBERT-256	0.942 ± 0.009	100.74%	0.125	1.870	79.85	**0.936** ± 0.002	**100.53**%	0.789	11.838	505.46
	TopicBERT-128	0.928 ± 0.015	99.251%	**0.107**	**1.610**	**68.76**	0.928 ± 0.002	99.677%	0.890	13.353	570.17
	TopicBERT-64	0.921 ± 0.006	98.502%	0.130	1.956	83.51	0.909 ± 0.015	97.636%	1.164	17.461	745.60
	Gain (performance)	↑ **1.604**%	-	-	-	-	↑ **0.537**%	-	-	-	-
	Gain (efficiency)	-	**99.251**%	↓**1.9**×	↓**1.9**×	↓**1.9**×	-	**100.53**%	↓**1.2**×	↓**1.2**×	↓**1.2**×
		20 Newsgroups (20NS) (news domain)					**Ohsumed** (medical domain)				
		F1	*Rtn*	T_{epoch}	T	CO_2	*F1*	*Rtn*	T_{epoch}	T	CO_2
baselines	*CNN*	0.786 ± 0.000	95.504%	0.109	1.751	74.76	0.684 ± 0.000	89.179%	0.177	7.090	302.74
	BERT-Avg	0.692 ± 0.000	84.083%	-	0.037	1.58	0.453 ± 0.000	59.061%	-	0.094	4.01
	BERT-Avg + DTR	0.731 ± 0.000	88.821%	-	0.051	2.18	0.543 ± 0.000	70.795%	-	0.191	8.16
	DistilBERT	0.816 ± 0.005	99.149%	**0.313**	**4.700**	**200.69**	0.751 ± 0.006	97.913%	**0.684**	**10.267**	**438.4**
	BERT	0.823 ± 0.007	100.00%	0.495	7.430	317.28	0.767 ± 0.002	100.00%	1.096	16.442	702.07
proposal	*TopicBERT-512*	0.826 ± 0.004	100.36%	0.507	7.606	324.76	**0.769** ± 0.005	**100.26**%	1.069	16.036	684.75
	TopicBERT-256	0.823 ± 0.016	100.00%	0.400	5.993	255.90	0.761 ± 0.001	99.217%	0.902	13.530	577.73
	TopicBERT-128	0.826 ± 0.004	100.36%	0.444	6.666	284.64	0.739 ± 0.006	96.349%	1.003	15.047	642.50
	TopicBERT-64	**0.830** ± 0.002	**100.85**%	0.605	9.079	387.66	0.711 ± 0.003	92.698%	1.334	20.008	854.34
	Gain (performance)	↑ **0.850**%	-	-	-	-	↑ **0.260**%	-	-	-	-
	Gain (efficiency)	-	**100.00**%	↓**1.2**×	↓**1.2**×	↓**1.2**×	-	**99.217**%	↓**1.2**×	↓**1.2**×	↓**1.2**×

Table 3: *TopicBERT* for document classification (macro-F1). *Rtn*: Retention in *F1* vs *BERT*; T_{epoch}: average epoch time (in hours); T: $T_{epoch} \times 15$ epochs; CO_2: 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*

tensions as: *TopicBERT* vs *BERT* (below) and *TopicDistilBERT* vs *DistilBERT* (in *supplementary*).

Experimental setup: For *BERT* component, we split the input sequence $\mathcal{D}$ 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 $\mathcal{D}$. 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, *NVDM* 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 (CO_2) 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 $1.9\times$ ($1.9\times$ reduction in CO_2) in fine-tuning while retaining (Rtn) 99.25% of *F1* of *BERT*. For IMDB and 20NS, *TopicBERT-256* reports similar performance to *BERT*, however with a speedup of $1.2\times$ 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* achieves 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 $1.4\times$ ($\sim 40\%$) and thus, a reduction of $\sim 40\%$ in CO_2 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-*

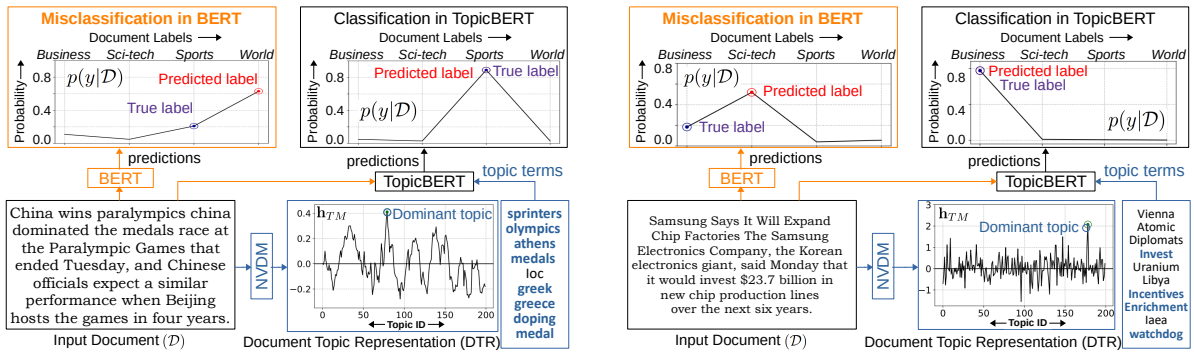

Figure 3: Interpretability analysis of document classification for AGnews dataset (for 2 different input documents): Illustration of document misclassification by *BERT* and correct classification by *TopicBERT* explained by the top key terms of dominant topic in DTR.

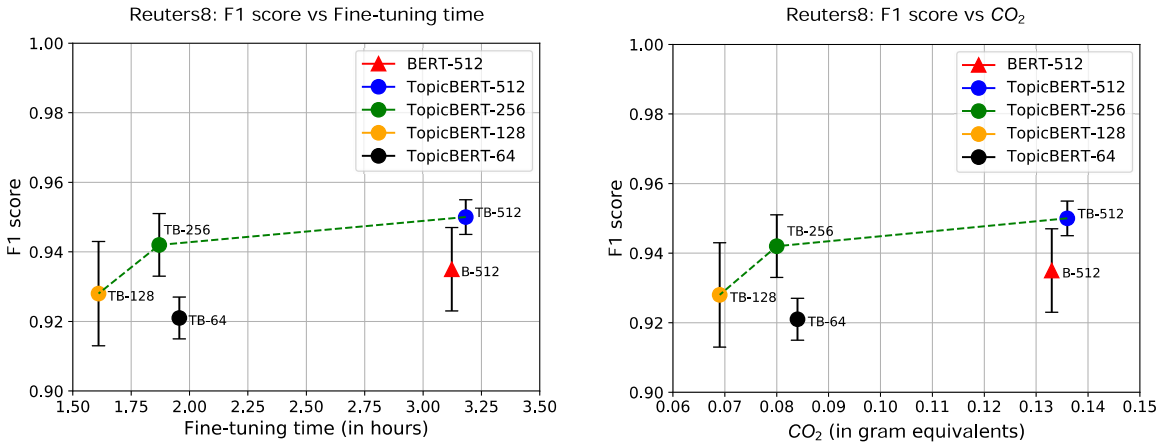

Figure 4: Pareto frontier analysis for Reuters8 dataset: *F1* score vs Fine-tuning time (left) and F1 score vs CO_2 (carbon footprint) (right). Here green dashed line represents Pareto frontier connecting optimal solutions

BERT) and its energy-efficient variants.

Analysis (Interpretability): For two different input documents, Figure 3 illustrates the misclassification by *BERT* and correct classification with explanation by *TopicBERT*, suggesting that the DTR ($\mathbf{h}_{TM}$ of *NVDM*) improves document understanding. The *TopicBERT* extracts key terms of the dominant topic (out of 200) discovered by the *NVDM* component for each document. Observe that the topic terms explain the correct classification in each case. (See *supplementary* for additional details and examples)

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 (right) to find the optimal solution that balances both fronts. Observe that the *TopicBERT*-512 outperforms all other *TopicBERT* variants and *BERT* baseline (B-512) in terms of *performance* i.e., *F1* score. However, *TopicBERT*-256 outperforms *BERT*-512 in terms of both, *performance* (*F1* score) and *efficiency* (Fine-tuning time/CO_2). Therefore, *TopicBERT*-256 represents the optimal solution with optimal sequence length of 256 for Reuters8 dataset.

4 Conclusion

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 (bmwi.de), grant 01MD19003E (PLASS (plass.io)) at Siemens AG - CT Machine Intelligence, Munich Germany.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Pankaj Gupta, Yatin Chaudhary, Florian Buettner, and Hinrich Schütze. 2019. Document informed neural autoregressive topic models with distributional prior. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6505–6512. AAAI Press.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel S. Weld. 2019. BERT for coreference resolution: Baselines and analysis. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5802–5807. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *CoRR*, abs/1910.09700.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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. *CoRR*, abs/1907.11692.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1727–1736. JMLR.org.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 14014–14024.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019a. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019b. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3645–3650. Association for Computational Linguistics.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from BERT into simple neural networks. *CoRR*, abs/1903.12136.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5754–5764.

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 (Reusters8-sized data) of maximum 512 sequence length = 12 hours on the hardware-type used

Therefore, using equation 1 in main paper,

CO_2 estimate in fine-tuning BERT = 0.07 × (5532 × 4 × 5) × 12 × 0.61 kg eq. = 56,692 × 2,20462 lbs eq = 124,985 lbs eq.

A.2 Data statistics and preprocessing

Table 4 shows data statistics of 5 datasets used in complementary + finetuning evaluation of our proposed *TopicBERT* model via Document Classification task. 20Newsgroups (20NS), Reuters8, AGnews are *news* domain datasets, whereas Imdb and Ohsumed datasets belong to *sentiment* and *medical* domains respectively. For NVDM component, we preprocess each dataset and extract vocabulary Z as follows: (a) tokenize documents into words, (b) lowercase all words, (d) remove stop words[2], and (c) remove words with frequency less than F_{min}. Here, $F_{min} = 100$ for large datasets i.e., Imdb, 20NS, AGnews and Ohsumed, whereas $F_{min} = 10$ for Reuters8 which is a small dataset.

[2]we use NLTK tool to remove stopwords

Dataset	**Train #docs**	**Dev #docs**	**Test #docs**	Z	L	N	b
Reuters8	4.9k	0.5k	2.1k	4813	8	512	4
Imdb	20k	5k	25k	6823	2	512	4
20NS	9.9k	1k	7.4k	4138	20	512	4
AGNews	118k	2k	7.6k	5001	4	128	32
Ohsumed†	24k	3k	2.9k	4553	20	512	4

Table 4: Preprocessed data statistics: **#docs** → number of documents, k → thousand, Z → vocabulary size of NVDM, L → total number of unique labels, N → sequence length used for *BERT* fine-tuning, b → batch-size used for BERT fine-tuning, (†) → multi-labeled dataset

Hyperparameter	Value(s)
Learning rate	0.001, 0.05
Hidden size (H)	256, 128
Batch size (b)	4, 32
Non-linearity (g)	sigmoid
Sampling frequency of $\mathbf{h}_{TM}$	5, 10
Number of topics (K)	50, 100,200

Table 5: Hyperparameters search and optimal settings for NVDM component of *TopicBERT* used in the experimental setup for document classification task.

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:

$$\mathcal{L}_{TopicBERT} = \alpha \log p(y = y_l|\mathcal{D}) + (1 - \alpha)\mathcal{L}_{NVDM}$$

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-*

	Models	Reuters8 (news domain)					20NS (news domain)				
		F1	*Rtn*	T_{epoch}	T	CO_2	*F1*	*Rtn*	T_{epoch}	T	CO_2
baselines	*CNN*	0.852 ± 0.000	91.123%	0.007	0.340	14.51	0.786 ± 0.000	95.504%	0.109	1.751	74.76
	DistilBERT	0.934 ± 0.003	100.00%	0.129	1.938	82.75	0.816 ± 0.005	100.000%	0.313	4.700	200.69
proposal	*TopicDistilBERT-512*	0.941 ± 0.007	100.75%	0.132	1.976	84.37	**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	98.284%	**0.190**	**2.850**	**121.69**
	TopicDistilBERT-128	0.911 ± 0.012	97.573%	0.096	1.444	61.66	0.797 ± 0.000	97.671%	0.387	5.800	247.66
	Gain (performance)	↑ **0.964%**	-	-	-	-	↑ **0.490%**	-	-	-	-
	Gain (efficiency)	-	**100.96%**	↓**1.5**×	↓**1.5**×	↓**1.5**×	-	**98.284%**	↓**1.6**×	↓**1.6**×	↓**1.6**×

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; CO_2: 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*

Hyperparameter	Value(s)
Learning rate*	2e-5
Hidden size (H_B)	768
Batch size (b)	[4, 32]
Non-linearity*	gelu
Maximum sequence length (N)	[512, 256, 128, 64, 32‡]
Number of attention heads*	12
Number of encoder layers* (n_l)	12
Vocabulary size*	28996
Dropout probability*	0.1
α	[0.1, 0.5, 0.9]

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

	Models	AGnews				
		F1	Rtn	T_{epoch}	T	CO_2
baselines	*CNN*	0.916 ± 0.000	97.447%	0.131	0.921	393.25
	BERT-Avg	0.903 ± 0.000	96.064%	-	0.075	3.20
	BERT-Avg + DTR	0.913 ± 0.000	97.128%	-	0.105	4.48
	DistilBERT-x	0.941 ± 0.001	100.10%	**0.491**	**7.361**	**314.31**
	BERT-x	0.940 ± 0.001	100.00%	0.952	14.281	609.80
proposal	*TopicBERT-128*	0.942 ± 0.003	100.21%	1.004	15.065	643.27
	TopicBERT-64	**0.943** ± 0.002	**100.31%**	0.723	10.838	462.78
	TopicBERT-32	0.938 ± 0.001	99.78%	0.846	12.688	541.66
	Gain (performance)	↑ **0.319** %	-	-	-	-
	Gain (efficiency)	-	**100.31%**	↓ **1.3** ×	↓ **1.3** ×	↓ **1.3** ×

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; CO_2: 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*

mance) compared to *BERT*, and (b) a significant speedup of 1.3× over *BERT* while retaining (*Rtn*) 100% of *F1* (*performance*) of *BERT* at the same time. This gain arises due to the improved document understanding using complementary topical semantics, via NVDM, in *TopicBERT* and its energy efficient versions.

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 1.5× (1.5×, i.e., ∼50% reduction in CO_2) 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 CO_2).

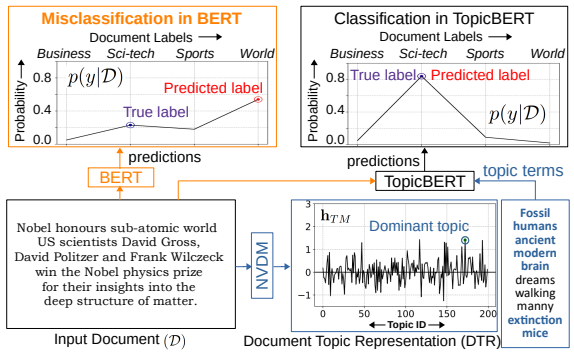

Figure 5: Interpretability analysis of document classification for AGnews dataset (for 2 input documents): Illustration of document misclassification by *BERT* model and correct classification by *TopicBERT* explained by the top key terms of dominant topic in DTR.

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.