Carburacy: Summarization Models Tuning and Comparison in Eco-Sustainable Regimes with a Novel Carbon-Aware Accuracy

Gianluca Moro[1, 2], **Luca Ragazzi**[1], **Lorenzo Valgimigli**[1]

[1]Department of Computer Science and Engineering, University of Bologna, Cesena Campus
Via dell'Università 50, I-47522 Cesena, Italy
[2]CNIT
{gianluca.moro, l.ragazzi, lorenzo.valgimigli}@unibo.it

Abstract

Generative transformer-based models have reached cutting-edge performance in long document summarization. Nevertheless, this task is witnessing a paradigm shift in developing ever-increasingly computationally-hungry solutions, focusing on effectiveness while ignoring the economic, environmental, and social costs of yielding such results. Accordingly, such extensive resources impact climate change and raise barriers to small and medium organizations distinguished by low-resource regimes of hardware and data. As a result, this unsustainable trend has lifted many concerns in the community, which directs the primary efforts on the proposal of tools to monitor models' energy costs. Despite their importance, no evaluation measure considering models' eco-sustainability exists yet. In this work, we propose `Carburacy`, the first carbon-aware accuracy measure that captures both model effectiveness and eco-sustainability. We perform a comprehensive benchmark for long document summarization, comparing multiple state-of-the-art quadratic and linear transformers on several datasets under eco-sustainable regimes. Finally, thanks to `Carburacy`, we found optimal combinations of hyperparameters that let models be competitive in effectiveness with significantly lower costs.

Introduction

In the last few years, we have witnessed remarkable progress on a broad range of natural language processing (NLP) tasks (Borgeaud et al. 2022; Moro et al. 2022; Frisoni et al. 2023), accomplished by increasingly large and computationally-hungry transformer-based models (Floridi and Chiriatti 2020; Smith et al. 2022) backed by high-performance dedicated hardware such as GPUs and TPUs (Wang, Wei, and Brooks 2019). Such solutions are proposed to obtain new state-of-the-art results, namely enhancing the score of automatic evaluation measures that focus on effectiveness (e.g., `ROUGE` (Lin 2004) for text summarization and `F1` for question answering), completely overlooking the economic, environmental, and social costs of attaining those outcomes (Strubell, Ganesh, and McCallum 2020). Thus, the AI community started raising concerns for this unsustainable trend (Bender et al. 2021), proposing tools to monitor energy costs without concrete solutions (Henderson et al. 2020).

An example of an NLP task affected by this shift is long document summarization, which aims to compress a lengthy input text into a shorter version while preserving the salient details in terms of informativeness and factual consistency (Koh et al. 2023). This task is dominated by large solutions (Wu et al. 2021; Mao et al. 2022) requiring massive labeled training data and high-memory GPUs (up to 48 GB) to obtain state-of-the-art results at the expense of a high carbon footprint (Dhar 2020; Patterson et al. 2022). Consequently, this untenable tendency narrows the research to large companies, raising barriers to small and medium organizations distinguished by low-resource regimes (e.g., GPUs of 12 GB memory and 100 training samples). Besides, in the face of the climate change problem, it is compulsory to minimize carbon sources to avoid rising global temperatures (Pörtner et al. 2022). Hence, we argue that the research must move toward environmental-friendly cutting-edge approaches (Tambe et al. 2021; Du et al. 2022; Moro and Ragazzi 2022; Frisoni et al. 2022b; Moro et al. 2023). Nevertheless, there is no automatic evaluation measure to capture the eco-sustainability of models, restricting the rise of new research directions toward greener solutions.

This paper presents `Carburacy`,[1] the first carbon-aware accuracy measure to evaluate models by considering the performance and the CO_2 emissions demanded to fulfill their effectiveness score. The key features of `Carburacy` are: (1) Applicability in multiple NLP tasks, namely those assessed by metrics that produce a score $\in [0, 1]$ (e.g., text summarization and classification). (2) Two hyperparameters used to tune what should be prioritized and rewarded (i.e., model effectiveness or eco-sustainability). To test `Carburacy`, we benchmark multiple state-of-the-art quadratic and linear transformers on various long document summarization datasets under challenging scenarios of limited hardware and data. Precisely, we study the impact of many hyperparameters on model effectiveness and eco-sustainability, optimizing the pipelines toward greener training. Moreover, we investigate the inference phase, which is paramount for model deployment at the business level.

Contributions. Our contributions are the following:
(1) *A novel carbon-aware accuracy measure to automatically consider eco-sustainability*. In this work, we propose

[1]The code is at https://github.com/disi-unibo-nlp/carburacy

Carburacy, which fits many NLP tasks and assesses models based on the cost needed to achieve their effectiveness. (2) *In-depth comparison of models in long document summarization under low-resource regimes*. We perform a deep benchmark of state-of-the-art models in long document summarization under challenging settings of hardware and data. (3) *Findings of optimal combinations of hyperparameters*. Thanks to Carburacy, we uncover the hyperparameters that let models obtain high effectiveness with low costs.

Related Work

Carbon Emissions Measures. Several works presented tools to estimate the energy consumption (Yang et al. 2017) and carbon footprint (Anthony, Kanding, and Selvan 2020; Naidu et al. 2021) of machine learning models and transformers (Cao et al. 2021; Lal et al. 2021), fighting against the problem of climate change by promoting future directions toward energy-efficient solutions. Despite their importance, no single evaluation measure considering the effectiveness and cost has been proposed, confining the development of new environmentally friendly state-of-the-art models.

Energy Costs Comparative Studies. Various works proposed studies on the environmental costs of models, ranging from computer vision (Li et al. 2016; Hampau et al. 2022) to NLP (Cao, Balasubramanian, and Balasubramanian 2020; Strubell, Ganesh, and McCallum 2020; Bannour et al. 2021; Zhou et al. 2021). Nonetheless, there is no comparative analysis of models in long document summarization and under low-resource scenarios, ignoring the many different infrastructures that research teams and organizations may have.

Our Work. Unlike prior contributions, our paper pioneers the exploration of the first carbon-aware accuracy measure (Carburacy), presenting an in-depth comparative analysis of cutting-edge models in long document summarization under low-resource regimes of hardware and data. Accordingly, Carburacy is not directly comparable with existing cost estimation metrics because it is the first measure that integrates both costs and effectiveness in a single score.

Methodology

In this section, we define the concept of effectiveness and cost and introduce our novel measure dubbed Carburacy.

Effectiveness

The effectiveness ($\mathcal{R}$ henceforth) is the result in terms of the model's accuracy. In long document summarization, $\mathcal{R}$ is evaluated with ROUGE (Lin 2004), which reckons the lexical overlaps of words and sentences between the inferred and target summary with three scores (i.e., r_1, r_2, r_L). For text summarization, we formally define $\mathcal{R} \in [0, 1]$ as:

$$\mathcal{R} = \frac{\mathcal{A}(r_1, r_2, r_L)}{1 + \sigma_r^2} \tag{1}$$

where $\mathcal{A}$ is the average function and σ_r^2 is the variance of the ROUGE scores. By dividing the averaged score by the variance, two models with the same average but different scores are not considered equal. Indeed, high variance means that the generated summary fails on some scores. For instance, our measure considers a model that obtains an $\{r_1, r_2, r_L\}$ score of $\{0.4, 0.2, 0.3\}$ better than a model that achieves $\{0.5, 0.15, 0.25\}$, despite having the same averaged score.

Eq. 1 explicitly defines the effectiveness of those tasks estimated with ROUGE, but it can be formulated for numerous NLP tasks (Table 1), e.g., text classification (Moro et al. 2018) and information retrieval (Moro and Valgimigli 2021).

Task	Eval Metric	Effectiveness ($\mathcal{R}$)
Summarization	ROUGE (r_1, r_2, r_L)	$\frac{\mathcal{A}(r_1, r_2, r_L)}{1 + \sigma_r^2}$
Translation	BLEU	BLEU
Classification	Accuracy / F1	Acc / F1 / $\frac{\mathcal{A}(\text{Acc}, \text{F1})}{1 + \sigma^2_{\text{Acc,F1}}}$

Table 1: Effectiveness formulas for different NLP tasks. Some tasks miss a custom formula because they are already evaluated with a single metric that produces a score $\in [0, 1]$.

Cost

The cost ($\mathcal{C}$ henceforth) denotes the resources required for a model to obtain $\mathcal{R}$. A first cost formulation is defined in Schwartz et al. (2020) as $\mathcal{C} \propto E \cdot D \cdot H$, where E is the cost of processing a single example, D is the size of the dataset, and H is the number of experiments for hyperparameter search.

Since we aim to investigate the correlation between the model effectiveness and cost by varying the hyperparameters, and the cost depends on the single instance's energy consumption for each sample of the dataset, we define $\mathcal{C}$ as:

$$\mathcal{C} = E \cdot D \tag{2}$$

Thereby, we want to determine how to gauge E formally. Since it is a common practice to consider only the computing device's dynamic energy consumption for comparing models (Gupta et al. 2022; Wu et al. 2022), although it does not represent the total environmental cost, we assume the following candidate sources:

- **Hardware cost**: the monetary cost of acquiring the required hardware.
- **CO_2 emissions and energy cost**: the CO_2 produced or the electricity used.
- **GPU memory usage**: the GB of GPU memory needed.

Since energy and CO_2 emissions are proportional, according to the provider policy on renewables, we only employ the latter to estimate the cost. We also neglet (1) the hardware buying cost because it is a one-time purchase strictly related to market costs and (2) the GPU memory occupation because it is unrelated to the carbon footprint, but we still report it for each experiment. Therefore, we define E as:

$$E = \textsc{Carbon}(\mathcal{M}(x)) \tag{3}$$

where CARBON is the kg of CO_2 produced by a model $\mathcal{M}$ to process an instance x. More precisely, CARBON is calculated by multiplying the power consumed by the computational infrastructure (quantified as kilowatt-hours) and the

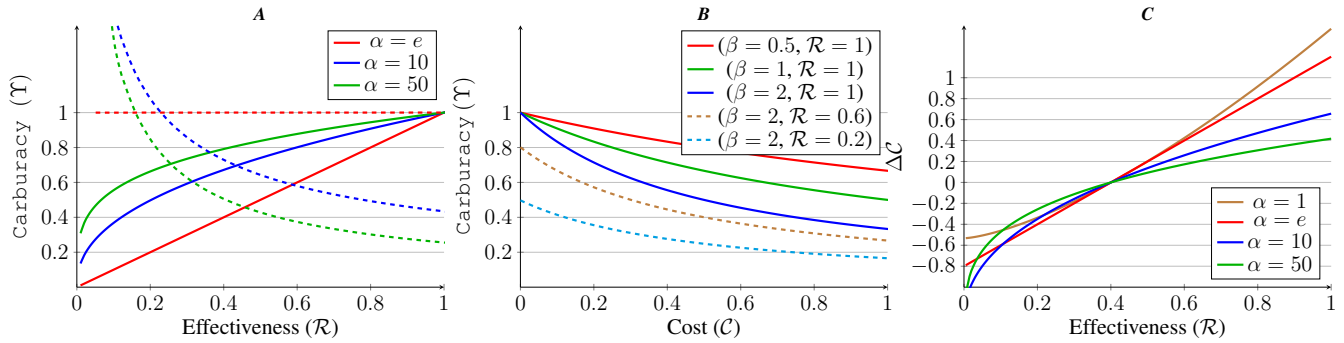

Figure 1: *A* analyzes how α impacts the `Carburacy` score ($\mathcal{C} = 0$); the dashed lines are the derivatives. *B* shows different behaviors by varying β and $\mathcal{R}$ ($\alpha = 10$). *C* reports the relation between $\Delta\mathcal{C}$, the difference of carbon emissions, and the effectiveness of keeping `Carburacy` stable using different α values, starting from a model with $\mathcal{R} = 0.4$ and $\Upsilon \cdot \beta = 0.5$.

carbon intensity of the electricity consumed for computation (quantified as kg of CO_2 per kilowatt-hour of electricity).

Carburacy

We present `Carburacy` (Υ henceforth),[2] the first carbon-aware accuracy measure, which represents the trade-off between the model effectiveness ($\mathcal{R}$) and cost ($\mathcal{C}$), as follows:

$$\Upsilon = \frac{e^{\log_\alpha \mathcal{R}}}{1 + \mathcal{C} \cdot \beta} \tag{4}$$

where $\alpha \geq e$ and $\beta > 0$ are hyperparameters to balance the effectiveness and the cost, respectively. Specifically, we adopt $e^{\log_\alpha \mathcal{R}}$ to simulate a non-linear trend of the effectiveness. As reported in Fig. 1-A, the function curvature is regulated by α, exhibiting a linear behavior for $\alpha = e$. Under this setting, α rewards the effectiveness gains that are mathematically more significant. For instance, the performance improvement $\{0.05 \rightarrow 0.10\}$ is mathematically more significant than $\{0.70 \rightarrow 0.75\}$ because the first doubles the effectiveness ($+100\%$) while the second is proportionally smaller ($+7.14\%$). Thus, the greater the α value, the greater the cost allowed for boosting lower performance (Fig. 1-A quantifies this reward by reporting the function's first derivative). Conversely, β weights the model cost (Fig. 1-B), assigning more or less importance to the carbon footprint over performance.

We define $\Delta\mathcal{C}$ as the emissions produced by improving the effectiveness without decreasing `Carburacy`:

$$\Delta\mathcal{C} = \frac{e^{\log_\alpha \mathcal{R}_n} - e^{\log_\alpha \mathcal{R}_o}}{\Upsilon \cdot \beta} \tag{5}$$

where $\mathcal{R}_n$ and $\mathcal{R}_o$ are the new and old effectiveness scores, respectively. As an example, given a model with $\mathcal{R} = 0.4$, and set $\beta = 1$ and $\Upsilon = 0.5$, Fig. 1-C shows the $\Delta\mathcal{C}$ value needed to keep stable Υ while decreasing/increasing $\mathcal{R}$ (several trends are shown based on different α values).

The decoding strategy at inference time plays a critical role in $\mathcal{R}$, but its cost is not yet considered in Eq. 4. For this reason, we measure `Carburacy` at training and inference time, also providing their harmonic mean:

$$\Upsilon_t = \frac{e^{\log_\alpha \mathcal{R}}}{1 + \mathcal{C}_t \cdot \beta_t} \qquad \Upsilon_i = \frac{e^{\log_\alpha \mathcal{R}}}{1 + \mathcal{C}_i \cdot \beta_i}$$
$$\Upsilon_m = 2 \cdot \frac{\Upsilon_t \cdot \Upsilon_i}{\Upsilon_t + \Upsilon_i} \tag{6}$$

where $\mathcal{C}_t, \beta_t$ and $\mathcal{C}_i, \beta_i$ are the cost and its modulator at training and inference time, respectively. In particular, while $\mathcal{C}_t$ represents the CO_2 emissions produced to execute the training phase with n samples, $\mathcal{C}_i$ considers only the cost of processing a single test instance. Indeed, datasets have test sets of a different number of samples which, unlike training, do not contribute to model effectiveness (i.e., a test set of $10,000$ instances is more expensive to process than a test set of 100 examples, but model effectiveness does not change). In addition, since the model is trained once and applied on demand, the inference cost, which is negligible w.r.t. the training, should be evaluated for a single instance.

In this study, we empirically set $\alpha = 10$, $\beta_t = 1$, and $\beta_i = 100$, but they can be tuned differently according to what should be prioritized (i.e., model effectiveness or eco-sustainability). In detail, we choose $\alpha = 10$ because it is a good trade-off between a flat trend and a too-steep curve (Fig. 1-A), $\beta_t = 1$ to represent the unit cost, and $\beta_i = 100$ to have the training and inference cost on similar scales.

Domain. $\{\, \Upsilon \in \mathbb{R} \mid 0 < \Upsilon \leq 1 \,\}$. We obtain that using the exponential and letting $\mathcal{R} \in [0, 1]$: for $\mathcal{R} = 0$, $\log_\alpha 0 \rightarrow -\infty$ and $e^{-\infty} \rightarrow 0^+$; for $\mathcal{R} = 1$, $\log_\alpha 1 = 0$ and $e^0 = 1$.

Dataset	Domain	# Docs	Source *# words*	Target *# words*
PUBMED	Biomedical	133,215	3224.4	214.4
ARXIV	Scientific	215,913	6913.8	292.8
GOVREPORT	Legal	19,466	9409.4	553.4

Table 2: Statistics of the datasets used as testbeds. The number of words are averaged across all instances.

[2] We assign the upsilon Greek letter (Υ) to `Carburacy` to promote the green AI concept since it resembles a tree.

Experiments

Datasets

We contemplate the following public datasets of different domains and text sizes as evaluation benchmarks (Table 2).[3]

- **PUBMED** (Cohan et al. 2018) consists of 133K biomedical papers from PubMed.
- **ARXIV** (Cohan et al. 2018) comprises 216K scientific papers from arXiv.org.
- **GOVREPORT** (Huang et al. 2021) includes 19K U.S. government reports.

Models

We compare the results of state-of-the-art quadratic and linear transformer-based models with a base and a large size, benchmarking 8 models overall.[4]

- **BART** (Lewis et al. 2020) is a transformer with a quadratic memory complexity in the input size limited to process sequences up to 1024 tokens in length due to the positional embedding mechanism.[5] We use the official `BART-base` and `BART-large` checkpoints.
- **T5** (Raffel et al. 2020) is a quadratic transformer without an input size limit, thanks to the relative embedding mechanism. We use the official 1.1 version of `T5-base` and `T5-large` pre-trained checkpoints.
- **LED** (Beltagy, Peters, and Cohan 2020) is a transformer with a linear memory and time complexity in the input size, thanks to sparse attention. We use the official `LED-base` and `LED-large` pre-trained checkpoints.
- **LONGT5** (Guo et al. 2022) is a linear transformer with sparse attention. We use the official `LongT5-base` and `LongT5-large` pre-trained checkpoints with the Transient-Global attention mechanism.

Experiment Setup

To compare state-of-the-art models in low-resource regimes, we simulate two real-world scenarios.

- **Limited GPU memory**: we benchmark models and their GPU usage by truncating input texts into different sizes, simulating a low-resource scenario of GPU memory in which the processing of long sequences throws "out of memory" exceptions (we apply input truncation because it is the general approach used when the input is longer than the GPU memory). Technically, we used the following sizes: 512, 1024, 2048, 4096, 8192, 16384.
- **Limited number of training instances**: we analyzed models' few-shot learning ability by fine-tuning them with a different number of training samples, simulating a low-resource scenario of labeled data. Concretely, we used the following instances: 1, 10, 100, 1000, 10000.

[3] All datasets are publicly available in Hugging Face: https://huggingface.co/datasets. We use the "*scientific_papers*" dataset for ARXIV and PUBMED and "*launch/gov_report*" for GOVREPORT.

[4] All pre-trained model checkpoints are publicly available in Hugging Face: https://huggingface.co/models

[5] More tokens could be fed into BART by training larger positional embeddings from scratch. However, the model is not pre-trained for such embeddings, resulting in unfair comparisons.

Implementation Details

We describe the implementation details of our experiments performed with PyTorch on a workstation using a single GPU NVIDIA RTX 3090 of 24 GB memory, 64 GB of RAM, and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz.

- **Training**: we use the first n samples of the training sets without shuffling, following the same approach in Chen and Shuai (2021). We train all models for 3 epochs, saving the model checkpoint that performed best on the validation sets. We apply gradient checkpointing to save memory, used the Adam optimizer, set the learning rate to $5e$-5, and set the seed to 42 for reproducibility.
- **Inference**: we evaluate all models on the first 100 instances of the test sets to save time. We use the beam search as the decoding strategy, setting the beam width to 2. We use an n-grams penalty of 3 for PUBMED and ARXIV and 5 for GOVREPORT.[6] We set the following output lengths (min-max) based on statistics in Table 2: PUBMED (100-300), ARXIV (150-350), GOVREPORT (500-1000). At the end of each training epoch, we use the same evaluation settings to monitor the performance on the first 10 samples of the validation sets (to simulate further a low-resource scenario), saving the checkpoint if $\mathcal{R}$ is higher than the performance of the previous epoch.

Metrics

We evaluated the models under several facets:

- **Effectiveness**: we used Eq. 1 to assess the effectiveness, considering the F1 scores for r_1, r_2, r_L. In detail, we split the summaries into sentences to compute the r_L score.[7]
- **Cost**: we used Eq. 2 to compute the models' cost. In particular, we leveraged the software CodeCarbon[8] to estimate E, namely the amount of CO_2 emissions produced by our infrastructure resources used to execute the model training and inference. Concretely, CodeCarbon monitors the CPU, TDP, and GPU energy consumption and converts the energy into CO_2 produced according to predefined conversion policy rules based on the local energy mix, defined in Lottick et al. (2019). Despite the importance of a complete vision of the overall footprint, we omit the cost of the pre-training phase of models for several reasons: (1) Pre-training is performed once and allows models to be fine-tuned in a wide range of downstream tasks by transferring and specializing knowledge. (2) We aim to compare multiple pre-trained models in the long document summarization task. (3) The footprint of the model pre-training is on higher scales than fine-tuning, making the latter irrelevant if not applying an empirical normalization. (4) The pre-training footprint is unessential at the business level because the application

[6] The penalty ensures that no n-gram appears twice by manually setting the probability of the following words that could create an already seen n-gram to 0.

[7] We used the `ROUGE` measure provided by NLG Metricverse (Frisoni et al. 2022a), using "*rougeLsum*" as r_L.

[8] https://codecarbon.io/

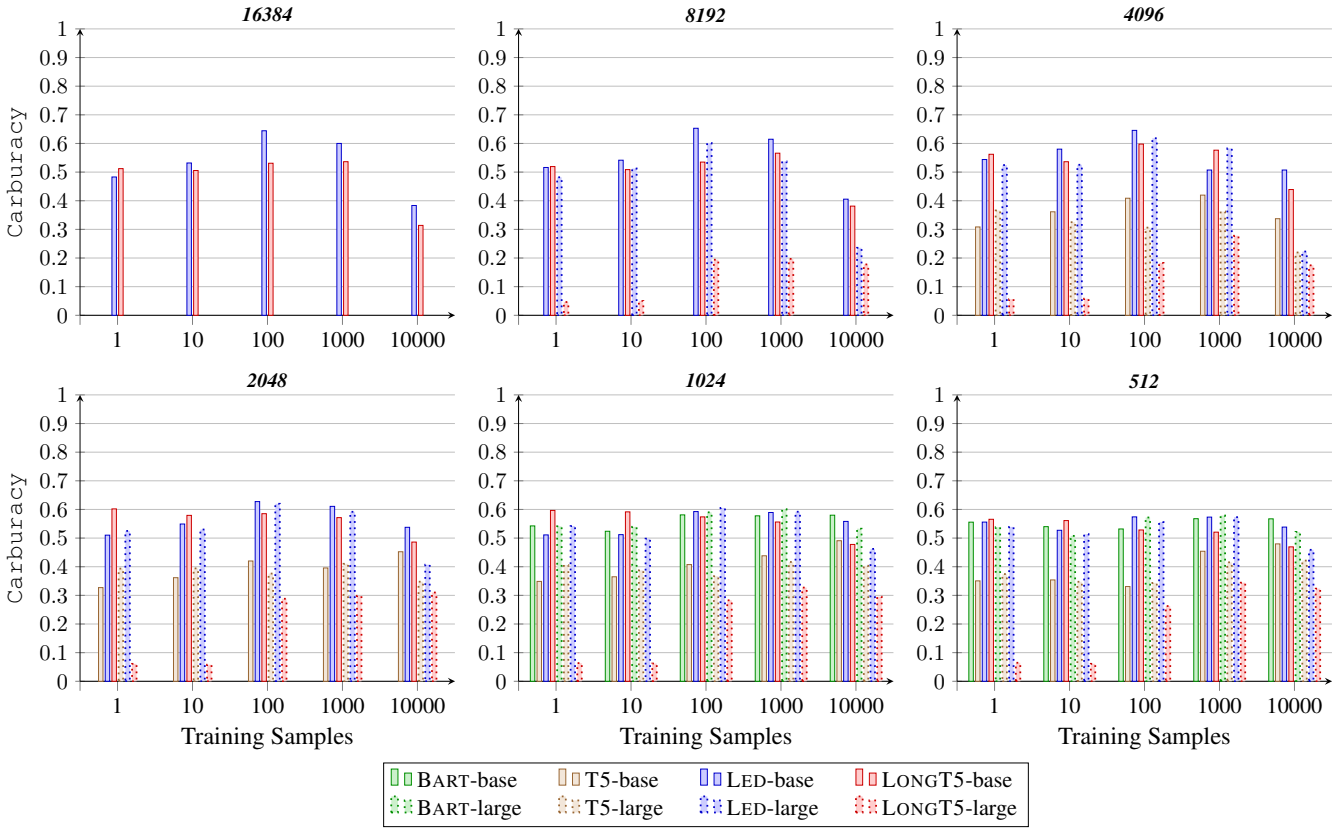

Figure 2: The Carburacy scores on GOVREPORT by varying the input size (reported on top) and the training samples.

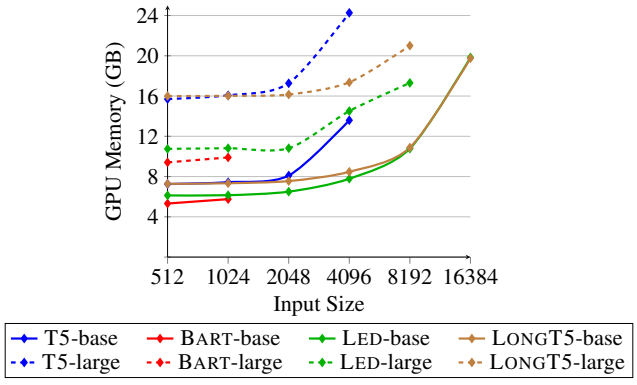

Figure 3: The GPU memory occupation on GOVREPORT by varying the input size at training time.

costs are related to energy consumption usage. (5) The CO_2 emissions for pre-training such models, required for the overall footprint, are generally not publicly released, making the re-pre-training from scratch the only option.

- **Efficiency**: we used our Carburacy measure (Eq. 6) to assess the models' efficiency. Thus, we refer to efficiency as the trade-off between effectiveness and costs.
- **Occupation**: we report the GB of GPU memory used at training time to determine the models' GPU occupation.

Results and Discussion

Analysis on the Number of Training Samples

Fig. 2 reports the Carburacy scores on GOVREPORT by feed models several input sizes and training instances (the GPU memory occupation at training time is shown in Fig. 3). The trend of Carburacy is not linear but follows a curve with a peak around 100 training documents (Table 5 reports some values for greater understanding). Similar behavior can be seen from almost all models tested on different settings and datasets (we show these results in the Appendix[9] for space reasons). This trend is due to the few-shot learning capabilities of language models, which achieve good results with meager costs. Indeed, the better the few-shot learning skills, the more eco-sustainable the models because they can achieve good effectiveness with minimal CO_2 emissions. With more training samples, the extra cost does not justify the slight improvement in effectiveness, thus leading to a drop in Carburacy. Large models, which contain more parameters, tend to obtain a similar Carburacy score to the base models, even if they produce significantly more CO_2; but, they are inefficient for long training sessions because they emit more CO_2 without significantly improving effectiveness. Indeed, with $10,000$ samples, base models achieve a better Carburacy score than their large versions.

[9] Appendix is at https://github.com/disi-unibo-nlp/carburacy

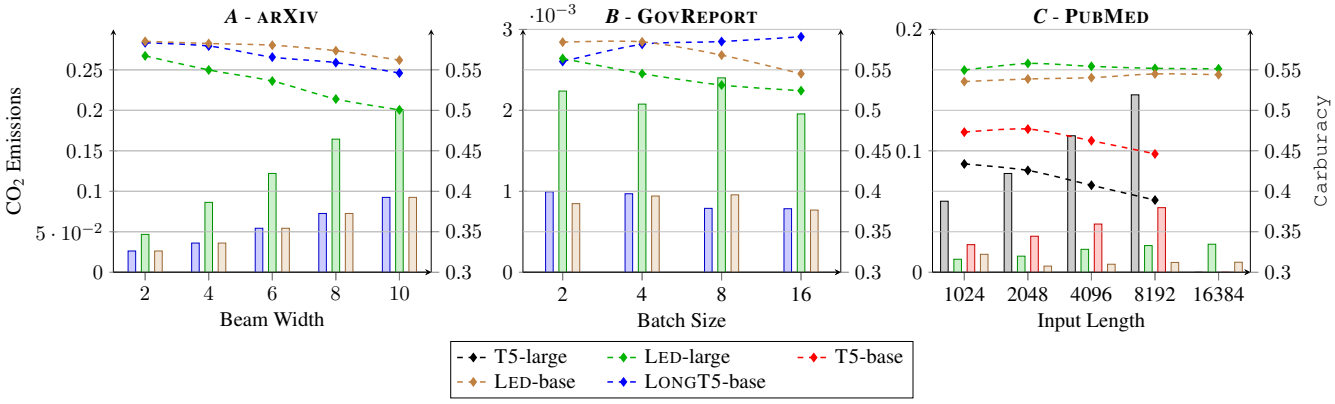

Figure 4: The graphs show the CO_2 (the bars) and Carburacy (dashed lines) for different models. *A* reports the decoding with different beam widths. *B* shows the training with different batch sizes. *C* reports the inference with different input lengths.

Analysis on the Input Size

Fig. 2 shows that enlarging the training input tokens results in effectiveness improvements more significant than the carbon emissions. Because of the nature of the task, which is long document summarization, extending the input tokens means giving the model more information. Thus, the same model in the same settings but with more input tokens generates a higher-quality summary. Conversely, a large number of input tokens makes the model consume more energy and GPU memory; however, the effectiveness rises significantly, increasing the overall Carburacy. The Carburacy score for extensive input lengths is, on average, better than for shorter ones. This trade-off makes linear transformers more eco-sustainable in long document summarization than models limited to processing few tokens, such as BART.

Analysis on the Training Batch Size

We study whether training with different batch sizes improves Carburacy. Specifically, Fig. 4-B shows the results by setting the following batch sizes: 1, 2, 4, 8, 16. We notice that enlarging the batch size reduces Carburacy slightly. This behavior is due to a drop in effectiveness while the cost remains stable. Indeed, the model learns during the backpropagation call while it adjusts the model weights to minimize the loss. Enlarging the batch size and keeping the number of training samples fixed leads to minor training steps with fewer backpropagation calls, resulting in a less-trained model. As is known, a large batch size is helpful to avoid overfitting and enhance generalization and optimal conversion, which are aspects concerning long training sessions. In our case, the training phase is relatively short, so the batch size keeps only the drawback of reducing the number of backpropagation calls, worsening the Carburacy score of models. The only exception among all models is LongT5-base, where Carburacy tends to improve.

Analysis on the Decoding Strategy

The decoding strategies used to generate the final summary are fundamental for the model's effectiveness (Wiher, Meister, and Cotterell 2022). Therefore, we investigate two crucial components in the decoding phase: the input size and the number of beams in the beam search decoding method. As reported in Table 3 and Table 4, the CO_2 emissions during the test phase are significantly larger than in training, even when the latter uses more samples. Indeed, the model's encoder and decoder are only called once at training time for each input example. On the contrary, the decoder is called $j \leq k \leq z$ times during inference, with j and z equal to the min and max number of target tokens to generate, respectively. Thus, investigating the CO_2 footprint of decoding techniques is crucial, particularly for business applications.

Beam Search. It is a decoding strategy that produces N different summaries for each document and keeps the most probable (N is the number of beams). Despite its widespread usage in text generation, it forces the model to work with multiple copies of the input, resulting in high energy consumption and GPU usage. Fig. 4-A contains the Carburacy score of three models using the following beam widths: 2, 4, 6, 8, 10. Results show that, for each model, Carburacy tends to decrease even when the effectiveness improves because the energy consumed is more than the performance gain. The picture also reports a linear trend of CO_2 emissions with respect to the number of beams.

Inference Input Length. In this experiment, we select two linear and quadratic models: LED-large, LED-base, T5-large, T5-base. All checkpoints are trained with 1024 as the input length on 100 training samples. We investigate different inference input lengths: 1024, 2048, 4096, 8192, 16384. Fig. 4-C reports how CO_2 emissions change according to the inference input length. Linear models keep a stable Carburacy while CO_2 emissions increase slightly, mitigated by a minor effectiveness improvement. Quadratic models fail, with a significant Carburacy drop due to the linear increase of CO_2 emissions. To sum up, increasing the input length at inference time does not help any model.

Findings

Q1: which are the most efficient models? Table 3 reports statistics of the top-3 efficient models for each dataset, which

Dataset	Model	*Hyperparameters* Input Size	Training Samples	*Effectiveness* ROUGE (R1 / R2 / RL)	$\mathcal{R}$	*Costs* kg CO_2 ($\mathcal{C}$) Train / Test	Memory (MB) Train / Test	*Efficiency* Carburacy (Υ_t / Υ_i / Υ_m)
PUBMED	LONGT5-B	2048	1000	39.46/14.83/35.76	29.69	0.0237 / 0.0212	7566 / 5516	55.08/**57.77**/**56.39**
	LED-L	8192	100	37.64/14.46/33.90	28.37	0.0073 / 0.0307	19540 / 11886	**56.62**/56.14/56.38
	LED-B	4096	100	35.76/12.96/32.22	26.98	0.0023 / 0.0090	8334 / 4366	56.21/55.87/55.93
ARXIV	LONGT5-B	2048	100	41.65/14.15/37.64	30.70	0.0025 / 0.0262	7734 / 6484	**59.44**/58.34/**58.88**
	LED-L	1024	100	41.08/13.22/37.29	30.07	0.0023 / 0.0157	10746 / 5704	58.93/58.43/58.68
	LED-B	8192	100	41.49/13.52/37.55	30.39	0.0042 / 0.0187	13150 / 6164	58.87/**58.52**/58.70
GOVREPORT	LED-B	8192	100	54.33/21.49/51.48	41.51	0.0036 / 0.0801	10776 / 5833	**67.54/63.20/65.30**
	LED-L	2048	100	50.40/17.70/47.37	37.67	0.0034 / 0.1021	10748 / 6254	64.79/59.38/61.97
	LONGT5-B	4096	100	50.40/17.31/47.36	37.52	0.0042 / 0.1729	8474 / 10872	64.52/55.70/59.78

Table 3: The results of the top-3 efficient base (B) and large (L) models for each dataset. Best scores are bolded for each dataset.

Model	Size	Samples	$\mathcal{R}$	$\mathcal{C}$ Train / Test	Υ_m
Quadratic					
T5-B	1024	1000	23.07	0.0101 / 0.0229	51.52
T5-L	512	1000	24.41	0.0185 / 0.0493	51.50
BART-B	512	1000	25.80	0.0033 / 0.0041	55.14
BART-L	512	100	24.80	0.0007 / 0.0085	54.29
Linear					
LONGT5-B	4096	100	25.64	0.0027 / 0.0359	54.18
LONGT5-L	8192	10000	25.07	0.7176 / 0.0710	25.96
LED-B	8192	10	25.88	0.0002 / 0.0096	55.32
LED-L	1024	100	25.84	0.0023 / 0.0107	55.07

Table 4: The comparison of base (B) and large (L) models with similar effectiveness on the PUBMED dataset.

Training Samples	$\mathcal{R}$	$\mathcal{C}$ Train / Test	Υ_t / Υ_i / Υ_m
1	26.90	0.0001 / 0.3416	56.52/42.14/48.28
10	29.85	0.0005 / 0.2230	59.06/48.37/53.18
100	42.17	0.0058 / 0.1160	**67.55**/61.59/**64.43**
1000	42.81	0.0633 / 0.1155	58.13/62.01/60.01
10000	45.67	0.5322 / 0.1146	27.40/**63.84**/38.34

Table 5: The results of LED-base with 16384 as input length on GOVREPORT by varying the number of training samples.

are LED-base, LED-large, and LONGT5-base. In Table 4, we analyze the models that achieve a similar $\mathcal{R}$ on PUBMED, finding that LED-base, LED-large, and BART-base are the three most efficient models. In general, linear models are more efficient than quadratic ones, even on short documents.

Q2: which are the most impacting factors? We found that the number of training samples and beams is directly proportional to the CO_2 emissions. A similar behavior regards input size at training and inference time for quadratic models. Differently, linear models do not suffer such drawbacks because raising the training input length leads to high effectiveness and `Carburacy`.

Q3: which is the most eco-sustainable pipeline? Based on our findings, we initially suggest training models with small batch sizes and raising them at the end to avoid overfitting. To reduce carbon emissions while maintaining good performance, we indicate using fewer training samples (e.g., 100) and small beam widths (e.g., 2) for ablation studies.

Model	$\mathcal{R}$	$\mathcal{C}$ Train / Test	Υ_m
Text Classification (IMDB dataset, 2011)			
DISTILBERT-base	89.64	0.0051 / 0.0012	95.12
BERT-base	89.99	0.0113 / 0.0021	94.98
Question Answering (SQUAD dataset, 2016)			
DISTILBERT-base	66.05	0.0048 / 0.0002	83.31
BERT-base	68.59	0.0080 / 0.0003	84.54

Table 6: The application of `Carburacy` to other NLP tasks.

Conclusion

We presented `Carburacy`, the first measure to score models by considering both effectiveness and costs. We compared multiple state-of-the-art transformers in long document summarization under real-world low-resource settings of hardware and data. Our results report the more environmental-friendly models and a set of hyperparameter combinations that let models achieve high effectiveness with low energy costs. Thanks to `Carburacy`, the research community can move toward carbon-aware state-of-the-art competitions (Table 6 shows the applicability of `Carburacy` to several NLP tasks). Future work includes extending our measure to general text mining (Frisoni and Moro 2020; Frisoni, Moro, and Carbonaro 2020), vision language (Moro and Salvatori 2022; Moro, Salvatori, and Frisoni 2023), non-NLP tasks (Lodi, Moro, and Sartori 2010; Cerroni et al. 2013), and ones characterized by unbounded evaluation metrics (i.e., with a score not in $[0, 1]$), such as pre-training with language modeling, a very energy-demanding session that deserves attention from an eco-sustainable point of view.

Acknowledgments

This research is partially supported by (i) the Complementary National Plan PNC-I.1, "Research initiatives for

innovative technologies and pathways in the health and welfare sector" D.D. 931 of 06/06/2022, DARE—DigitAl lifelong pRevEntion initiative, code PNC0000002, CUP B53C22006450001, (ii) the PNRR, M4C2, FAIR—Future Artificial Intelligence Research, Spoke 8 "Pervasive AI," funded by the European Commission under the NextGeneration EU program. We thank the Maggioli Group[10] for granting the Ph.D. scholarship to L. Ragazzi and L. Valgimigli.

References

Anthony, L. F. W.; Kanding, B.; and Selvan, R. 2020. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models. *CoRR*, abs/2007.03051.

Bannour, N.; Ghannay, S.; Névéol, A.; and Ligozat, A. 2021. Evaluating the carbon footprint of NLP methods: a survey and analysis of existing tools. In *SustaiNLP@EMNLP*, 11–21. ACL.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-Document Transformer. *CoRR*, abs/2004.05150.

Bender, E. M.; Gebru, T.; McMillan-Major, A.; and Shmitchell, S. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *FAccT*, 610–623. ACM.

Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; et al. 2022. Improving Language Models by Retrieving from Trillions of Tokens. In *ICML, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 2206–2240. PMLR.

Cao, Q.; Balasubramanian, A.; and Balasubramanian, N. 2020. Towards Accurate and Reliable Energy Measurement of NLP Models. In *SustaiNLP@EMNLP*, 141–148. ACL.

Cao, Q.; Lal, Y. K.; Trivedi, H.; Balasubramanian, A.; et al. 2021. IrEne: Interpretable Energy Prediction for Transformers. In *ACL/IJCNLP (1)*, 2145–2157. ACL.

Cerroni, W.; Moro, G.; Pirini, T.; and Ramilli, M. 2013. Peer-to-Peer Data Mining Classifiers for Decentralized Detection of Network Attacks. In *ADC*, volume 137 of *CRPIT*, 101–108. ACS.

Chen, Y.; and Shuai, H. 2021. Meta-Transfer Learning for Low-Resource Abstractive Summarization. In *AAAI*, 12692–12700. AAAI Press.

Cohan, A.; Dernoncourt, F.; Kim, D. S.; Bui, T.; et al. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *NAACL-HLT (2)*, 615–621. ACL.

Dhar, P. 2020. The carbon impact of artificial intelligence. *Nat. Mach. Intell.*, 2(8): 423–425.

Du, N.; Huang, Y.; Dai, A. M.; Tong, S.; et al. 2022. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, 5547–5569. PMLR.

Floridi, L.; and Chiriatti, M. 2020. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds Mach.*, 30(4): 681–694.

[10]https://www.maggioli.com/who-we-are/company-profile

Frisoni, G.; Carbonaro, A.; Moro, G.; Zammarchi, A.; and Avagnano, M. 2022a. NLG-Metricverse: An End-to-End Library for Evaluating Natural Language Generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, 3465–3479. Gyeongju, Republic of Korea: International Committee on Computational Linguistics.

Frisoni, G.; Italiani, P.; Salvatori, S.; and Moro, G. 2023. Cogito Ergo Summ: Abstractive Summarization of Biomedical Papers via Semantic Parsing Graphs and Consistency Rewards. In *AAAI*, 1–9. AAAI Press.

Frisoni, G.; Mizutani, M.; Moro, G.; and Valgimigli, L. 2022b. BioReader: a Retrieval-Enhanced Text-to-Text Transformer for Biomedical Literature. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 5770–5793. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Frisoni, G.; and Moro, G. 2020. Phenomena Explanation from Text: Unsupervised Learning of Interpretable and Statistically Significant Knowledge. In *Data Management Technologies and Applications - 9th International Conference, DATA 2020, Virtual Event, July 7-9, 2020, Revised Selected Papers*, volume 1446 of *Communications in Computer and Information Science*, 293–318. Springer.

Frisoni, G.; Moro, G.; and Carbonaro, A. 2020. Learning Interpretable and Statistically Significant Knowledge from Unlabeled Corpora of Social Text Messages: A Novel Methodology of Descriptive Text Mining. In *Proceedings of the 9th International Conference on Data Science, Technology and Applications, DATA 2020, Lieusaint, Paris, France, July 7-9, 2020*, 121–132. SciTePress.

Guo, M.; Ainslie, J.; Uthus, D. C.; Ontañón, S.; et al. 2022. LongT5: Efficient Text-To-Text Transformer for Long Sequences. In *NAACL-HLT (Findings)*, 724–736. ACL.

Gupta, U.; Kim, Y. G.; Lee, S.; Tse, J.; et al. 2022. Chasing Carbon: The Elusive Environmental Footprint of Computing. *IEEE Micro*, 42(4): 37–47.

Hampau, R.; Kaptein, M.; van Emden, R.; Rost, T.; et al. 2022. An empirical study on the Performance and Energy Consumption of AI Containerization Strategies for Computer-Vision Tasks on the Edge. In *EASE 2022, Gothenburg, Sweden, June 13 - 15, 2022*, 50–59. ACM.

Henderson, P.; Hu, J.; Romoff, J.; Brunskill, E.; et al. 2020. Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. *CoRR*, abs/2002.05651.

Huang, L.; Cao, S.; Parulian, N. N.; Ji, H.; et al. 2021. Efficient Attentions for Long Document Summarization. In *NAACL-HLT*, 1419–1436. ACL.

Koh, H. Y.; Ju, J.; Liu, M.; and Pan, S. 2023. An Empirical Survey on Long Document Summarization: Datasets, Models, and Metrics. *ACM Comput. Surv.*, 55(8): 154:1–154:35.

Lal, Y. K.; Singh, R.; Trivedi, H.; Cao, Q.; et al. 2021. IrEneviz: Visualizing Energy Consumption of Transformer Models. In *EMNLP (Demos)*, 251–258. ACL.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; et al. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*, 7871–7880. ACL.

Li, D.; Chen, X.; Becchi, M.; and Zong, Z. 2016. Evaluating the Energy Efficiency of Deep Convolutional Neural Networks on CPUs and GPUs. In *BDCloud-SocialCom-SustainCom*, 477–484. IEEE Computer Society.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: ACL.

Lodi, S.; Moro, G.; and Sartori, C. 2010. Distributed data clustering in multi-dimensional peer-to-peer networks. In *(ADC 2010), Brisbane, Australia, 18-22 January, 2010, Proceedings*, volume 104 of *CRPIT*, 171–178. Australian Computer Society.

Lottick, K.; Susai, S.; Friedler, S. A.; and Wilson, J. P. 2019. Energy Usage Reports: Environmental awareness as part of algorithmic accountability. *CoRR*, abs/1911.08354.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; et al. 2011. Learning Word Vectors for Sentiment Analysis. In *ACL, 19-24 June, 2011, Portland, Oregon, USA*, 142–150. ACL.

Mao, Z.; Wu, C. H.; Ni, A.; Zhang, Y.; et al. 2022. DYLE: Dynamic Latent Extraction for Abstractive Long-Input Summarization. In *ACL (1)*, 1687–1698. ACL.

Moro, G.; Pagliarani, A.; Pasolini, R.; and Sartori, C. 2018. Cross-domain & In-domain Sentiment Analysis with Memory-based Deep Neural Networks. In *Proceedings of IC3K 2018, Volume 1: KDIR, Seville, Spain, September 18-20, 2018*, 125–136. SciTePress.

Moro, G.; and Ragazzi, L. 2022. Semantic Self-Segmentation for Abstractive Summarization of Long Documents in Low-Resource Regimes. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual Event, February 22 - March 1, 2022*, 11085–11093. AAAI Press.

Moro, G.; Ragazzi, L.; Valgimigli, L.; and Freddi, D. 2022. Discriminative Marginalized Probabilistic Neural Method for Multi-Document Summarization of Medical Literature. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 180–189. Dublin, Ireland: ACL.

Moro, G.; Ragazzi, L.; Valgimigli, L.; Frisoni, G.; Sartori, C.; and Marfia, G. 2023. Efficient Memory-Enhanced Transformer for Long-Document Summarization in Low-Resource Regimes. *Sensors*, 23(7).

Moro, G.; and Salvatori, S. 2022. Deep Vision-Language Model for Efficient Multi-modal Similarity Search in Fashion Retrieval. In *SISAP 2022, Bologna, Italy, October 5-7, 2022, Proceedings*, volume 13590 of *Lecture Notes in Computer Science*, 40–53. Springer.

Moro, G.; Salvatori, S.; and Frisoni, G. 2023. Efficient Text-Image Semantic Search: a Multi-modal Vision-Language Approach for Fashion Retrieval. *Neurocomputing*.

Moro, G.; and Valgimigli, L. 2021. Efficient Self-Supervised Metric Information Retrieval: A Bibliography Based Method Applied to COVID Literature. *Sensors*, 21(19): 6430.

Naidu, R.; Diddee, H.; Mulay, A.; Vardhan, A.; et al. 2021. Towards Quantifying the Carbon Emissions of Differentially Private Machine Learning. *CoRR*, abs/2107.06946.

Patterson, D. A.; Gonzalez, J.; Hölzle, U.; Le, Q. V.; et al. 2022. The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink. *Computer*, 55(7): 18–28.

Pörtner, H. O.; Roberts, D. C.; Adams, H.; Adler, C.; et al. 2022. Climate change 2022: impacts, adaptation and vulnerability. *IPCC*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; et al. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2383–2392. ACL.

Schwartz, R.; Dodge, J.; Smith, N. A.; and Etzioni, O. 2020. Green AI. *Commun. ACM*, 63(12): 54–63.

Smith, S.; Patwary, M.; Norick, B.; LeGresley, P.; et al. 2022. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. *CoRR*, abs/2201.11990.

Strubell, E.; Ganesh, A.; and McCallum, A. 2020. Energy and Policy Considerations for Modern Deep Learning Research. In *The Thirty-Fourth AAAI 2020, New York, NY, USA, February 7-12, 2020*, 13693–13696. AAAI Press.

Tambe, T.; Hooper, C.; Pentecost, L.; Jia, T.; et al. 2021. EdgeBERT: Sentence-Level Energy Optimizations for Latency-Aware Multi-Task NLP Inference. In *MICRO*, 830–844. ACM.

Wang, Y.; Wei, G.; and Brooks, D. 2019. Benchmarking TPU, GPU, and CPU Platforms for Deep Learning. *CoRR*, abs/1907.10701.

Wiher, G.; Meister, C.; and Cotterell, R. 2022. On Decoding Strategies for Neural Text Generators. *CoRR*, abs/2203.15721.

Wu, C.; Raghavendra, R.; Gupta, U.; Acun, B.; et al. 2022. Sustainable AI: Environmental Implications, Challenges and Opportunities. In *MLSys*. mlsys.org.

Wu, J.; Ouyang, L.; Ziegler, D. M.; Stiennon, N.; et al. 2021. Recursively Summarizing Books with Human Feedback. *CoRR*, abs/2109.10862.

Yang, T.; Chen, Y.; Emer, J. S.; and Sze, V. 2017. A method to estimate the energy consumption of deep neural networks. In *ACSSC*, 1916–1920. IEEE.

Zhou, X.; Chen, Z.; Jin, X.; and Wang, W. Y. 2021. HULK: An Energy Efficiency Benchmark Platform for Responsible Natural Language Processing. In *EACL*, 329–336. ACL.