Reduce, Reuse, Recycle: Green Information Retrieval Research

Harrisen Scells
The University of Queensland
Brisbane, Australia
h.scells@uq.edu.au

Shengyao Zhuang
The University of Queensland
Brisbane, Australia
s.zhuang@uq.edu.au

Guido Zuccon
The University of Queensland
Brisbane, Australia
g.zuccon@uq.edu.au

ABSTRACT

Recent advances in Information Retrieval utilise energy-intensive hardware to produce state-of-the-art results. In areas of research highly related to Information Retrieval, such as Natural Language Processing and Machine Learning, there have been efforts to quantify and reduce the power and emissions produced by methods that depend on such hardware. Research that is conscious of the environmental impacts of its experimentation and takes steps to mitigate some of these impacts is considered 'Green'. Given the continuous demand for more data and power-hungry techniques, Green research is likely to become more important within the broader research community. Therefore, within the Information Retrieval community, the consequences of non-Green (in other words, Red) research should at least be considered and acknowledged. As such, the aims of this perspective paper are fourfold: (1) to review the Green literature not only for Information Retrieval but also for related domains in order to identify transferable Green techniques; (2) to provide measures for quantifying the power usage and emissions of Information Retrieval research; (3) to report the power usage and emission impacts for various current IR methods; and (4) to provide a framework to guide Green Information Retrieval research, taking inspiration from 'reduce, reuse, recycle' waste management campaigns, including salient examples from the literature that implement these concepts.

CCS CONCEPTS

• **Information systems** → *Retrieval efficiency*; • **Hardware** → *Impact on the environment*.

KEYWORDS

Green IR, Deep Learning, Efficiency, Emissions

ACM Reference Format:
Harrisen Scells, Shengyao Zhuang, and Guido Zuccon. 2022. Reduce, Reuse, Recycle: Green Information Retrieval Research. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3477495.3531766

1 INTRODUCTION

Recently, impressive progress has been made with Information Retrieval (IR) methods based on neural networks and large language models [24, 88, 94]. However, despite the achievements of these models, one downside is the high energy costs required to train and use them in production. Indeed, these methods often require financial and environmental costs, primarily due to the dependence on specialised hardware such as GPUs. In related fields such as Natural Language Processing (NLP), Machine Learning (ML), and the broad field of Artificial Intelligence (AI), discussions regarding the energy impact, and more importantly, emissions, of methods is increasing. One such study from Strubell et al. [77] notes that training a typical NLP pipeline produces more emissions than an average human produces each year in the U.S.A. and that training a large neural transformer model produces approximately five times more emissions than the average lifetime of a car including fuel. There have also been concerns about the emissions produced by computer systems more broadly [2, 4].

Firstly, consider that these numbers are for training a *single* model: often, experiments require several rounds of training because of bugs, hyperparameter tuning, or any number of other reasons. And then consider the number of papers *submitted* to conferences such as SIGIR (in the order of thousands). Finally, consider that most researchers will submit to multiple venues each year. This perspective paper aims to convince the IR community that the emissions produced through IR research and subsequent deployment through production can be considerable. We do this in four ways: (1) by reviewing the areas of research that have been explored in the past to address similar problems such as power efficiency, (2) by providing considerations for how the IR community can make their research 'Greener' through a practical framework, (3) by providing a measure that can be used to quantify the emissions for IR research and production IR systems, and (4) by performing experiments that demonstrate the potential emissions generated by typical IR research pipelines (i.e., cost of a result) and production systems (i.e., cost of search at different scales). With all of these aspects, our goal is to push the community to emphasise considering the environmental impacts of their research. We acknowledge that research into the energy efficiency in the field of IR is not new: environmentally sustainable IR has been of concern for at least a decade [13]. Although historically, the focus has been on utilising energy-efficient hardware or specialised scenarios such as distributed search. Instead, this perspective paper focuses on the current generation of data and power-hungry IR techniques and their reliance on costly specialised hardware. We believe that discussing the environmental cost of doing research with these techniques is becoming increasingly important. Moreover, as the authors of this paper, we sought to quantify the impact of our experiments and found no straightforward way to do so.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531766

In order for us to quantify the emissions of IR research, we can take two approaches: Life Cycle Assessment (LCA) [23] and power consumption measurement [77]. According to the ISO standard definition, LCA is the "*compilation and evaluation of the inputs, outputs and the potential environmental impacts of a product system throughout its life cycle*" [1]. Given the complexity and resource intensiveness of the LCA approach [13], most studies in related domains measure emissions using the second approach. In this perspective paper, we also take the second approach. Given that there is a reasonable amount of terminology associated with this approach, before continuing with the rest of the paper, we first introduce some common terminology that IR researchers and practitioners may not be familiar with.

2 TERMINOLOGY

Information Retrieval researchers and Computer Science practitioners, in general, may not be familiar with the nomenclatures used to discuss the themes of this paper. The key terms and the technical terminologies used to discuss them are listed below.

Energy and Power Lottick et al. [47] provide a succinct explanation of energy and power. They define **energy** as "*an amount of work done, or to, an object.*" Energy is measured in joules; the exact definition of how a joule is measured is not important to this paper. On the other hand, Lottick et al. [47] define **power** as "*the energy per unit time*". Power is measured in watts: one watt equals one joule per second. It is more convenient to use larger measurements such as kilowatts (kW, 1,000 watts) at larger scales. However, kW measures only the rate at which energy is used, not the total energy used. For this, a common unit of measure is the kilowatt-hour (kWh), which Lottick et al. [47] also provides a succinct definition as 'the energy consumed at a rate of one kilowatt for one hour'.

Emissions This is the unit by which we quantify a given IR experiment's impact on the environment. While CO_2 is the most common greenhouse gas, many other gasses or factors are often involved. For this reason, it is typical to measure the emissions produced by experiments as $kgCO_2e$, in other words, the amount (in kilograms) of CO_2 equivalent emissions. We provide a method for calculating this measure in Section 4.

Green IR This is a specific use of the word 'Green', which relates to the phrase 'Green AI' proposed by Schwartz et al. [72]. In their article, they refer to 'Green AI' as 'AI research that yields novel results while taking into account the computational cost, encouraging a reduction in resources spent.' Naturally, this is in contrast with 'Red AI', which Schwartz et al. refer to as 'AI research that seeks to improve accuracy (or related measures) through the use of massive computational power while disregarding the cost — essentially "buying" stronger results'. We believe that these concepts map meaningfully to IR in general, so we adapt the phrases of Schwartz et al. to apply to this community. Indeed, the concept of Green IR was initially introduced by Chowdhury [13] to refer specifically to the emissions produced by IR experiments and climate change in general. Our usage of Green IR extends this idea to a broader context that also encapsulates Red IR. One other facet to highlight when considering Green IR versus Red IR, which Schwartz et al. raise in their article, is that Red experiments are valuable in pushing the boundaries of a field, can promote future work in efficiency, and the costs may be amortised over time for methods that do not require retraining. This is to say that Red IR experiments should not be abandoned but that Green IR should be considered, at the very least.

3 LITERATURE REVIEW

At the time of writing, there has been a surge of research articles that aim to assess the emission intensity of computer science experimentation and research. We focus our efforts first on studies within fields related to IR, such as NLP and ML in general. This first focus is because there is a significant overlap between the methods used in IR and these related domains, especially with the uptake of deep neural network and transformer approaches that are currently popular research areas. We then turn our attention to efforts in IR that have sought to develop methods to reduce emissions or propose considerations for practitioners to consider when designing and executing experiments.

3.1 NLP and Machine Learning

3.1.1 Methods for Quantifying Emissions. One of the most prominent papers that aims to quantify the environmental impact of research is from Strubell et al. [77], who propose a framework for calculating the emissions of deep learning experiments in NLP. One critical limitation of that paper is that the authors only consider the energy usage for training a model. Training models account for only a relatively small amount of carbon emissions; Amazon estimates that 90% of the ML infrastructure costs are related to model inference [37]. In this paper, we build upon their framework to include the quantification of emissions for aspects of typical IR experimental pipelines, including the cost of indexing and querying.

3.1.2 Methods for Reducing Emissions. In addition to methods that seek to quantify the number of emissions produced by NLP and ML methods, several studies have recently sought to propose methods for reducing the number of emissions produced by experimentation and research. One of the most comprehensive papers on this topic is a survey on green deep learning by Xu et al. [89]. They define 'green deep learning' as using more energy-efficient architectures, training methods, inference methods, and data usage techniques. They identify too many methods to include as citations here; however, some papers to note that are highly relevant to IR include: (1) Naidu et al. [57], who quantify the emissions of differentially private Machine Learning algorithms and show that more robust privacy regimes lead to more emissions produced. This paper is highly relevant to federated learning, a new area of research in IR. Indeed, recent findings of federated learning have demonstrated that it produces fewer emissions than traditional GPU-based ML pipelines [68]; (2) Yusuf et al. [92], who quantify the emissions of machine translation and show that some pairs of languages produce more emissions to train than others. This paper is highly relevant to cross-lingual IR, which is an essential and highly-studied area of research; and (3) Wiesner et al. [85], who propose a method for reducing carbon emissions of Machine Learning experiments by scheduling the training of models during non-peak hours, and found that shifting workloads to the next day can reduce emissions by 5% regardless of the region. This paper is also relevant to IR as the uptake of deep learning models increases.

Name	CPU	DRAM	GPU	Network	Repository
CodeCarbon [71]	✓	✓	✓	✗	https://github.com/mlco2/codecarbon
pyJoules	✓	✓	✓	✗	https://github.com/powerapi-ng/pyJoules
energyusage [47]	✓	✓	✓	✗	https://github.com/responsibleproblemsolving/energy-usage
Carbontracker [3]	✓	✗	✓	✗	https://github.com/lfwa/carbontracker
Experiment Impact Tracker [33]	✓	✗	✓	✗	https://github.com/Breakend/experiment-impact-tracker
Cumulator [81]	✓	✓	✓	✓	https://github.com/epfl-iglobalhealth/cumulator

Table 1: Off-the-shelf software libraries that can be used to record some or all of the variables necessary for calculating the kgCO$_2$e/kWh of Information Retrieval experiments. The ✓and ✗symbols indicate whether these libraries are capable of recording the associated measurement.

3.1.3 Tools for Measuring Emissions. There have also been several tools that seek to estimate the emissions produced by ML experiments. Some of these tools are highlighted later in Table 1. These tools [3, 33, 47, 71, 81] directly measure power draw in Watts using recently implemented hardware APIs in certain CPUs and GPUs, and in some cases, provide estimates of network power usage too. Another set of tools that can make less accurate estimates of emissions produced by ML experiments simply do so based on the kind of hardware used and how long experiments were run for [41].

3.2 Information Retrieval

3.2.1 Green IR in the Literature. The concept of Green IR is indeed not new. Several papers have been published that discuss the environmental considerations of IR methods. For example, to the best of our knowledge, the first to propose an agenda for Green IR was Chowdhury [14, 15] who focus their efforts on measuring and reducing emissions within a digital libraries context.

3.2.2 Energy and Power Usage. In addition, there have also been several papers that put forward methods to reduce the energy and power of IR experiments. One prominent name in this space is Catena, who proposes methods to measure the energy consumption of querying search engines [8, 11], to consider power management of searching web search engines [10], methods for energy-efficient query processing in web search engines [12], managing the energy usage of distributed web search [9], and measuring the costs of multi-center web search engines [6]. The focus of these studies are on more traditional IR experimental pipelines and do not consider current methods that exploit GPUs. However, such methods can still be used today, especially for early stages in retrieval pipelines, prior to more expensive operations like top-k re-ranking.

3.2.3 Efficiency. There is a long and rich history of efficiency in IR research [17, 82, 86]. Algorithms that are space- and time-efficient naturally fit into the Green IR category. However, in addition to the space and time measurements typically recorded (as well as the effectiveness trade-offs, if any), the measurements of power usage offer another interesting dimension of analysis that can be further used to contrast with other efficiency measures. Indeed, while some energy-efficiency focused studies have already been conducted [10, 25]. and such analysis of effectiveness trade-offs provides a clear direction for future work.

3.2.4 Neural IR and Power Usage. One highly successful recent trend in Information Retrieval is the use of transformer language models [83] such as BERT [22] that make heavy use of GPUs for both training and inference [21, 24, 35, 44, 49, 52, 69, 88, 94, 95]. To date, there have been no IR-focused studies to investigate the power usage of such models. Although some studies have focused on improving the efficiency of the models (i.e., reducing the inference time for ranking), this is not the same as power usage. An efficient neural model may still use more power than an efficient CPU-based method even if they run in the same amount of time. However, there is a hidden cost associated with measuring efficiency through running time, or latency, which is that often GPU-based experimentation is highly parallelised. In these cases, latency only captures the longest path through a parallelised job, not the cumulative amount of work expended, which is what we aim to address. In this paper, we study the power usage of these new models and compare them to more traditional IR pipelines. The distinction between efficiency and power usage is important, and we believe a key but underexplored area of research in current neural IR trends.

4 QUANTIFYING GREEN IR

We believe that it is important to quantify the power usage and emissions impact of IR research, especially given the power utilisation of recent advancements in GPU-powered IR research. As of writing, there are already several practical methods and frameworks for quantifying the environmental impact of machine learning models. As the following measures provide an accurate estimation for the power and emissions produced by experimentation, they are ideal for quantifying the impact of emissions. The following measures are used within this paper to compare several typical IR pipelines.

Strubell et al. [77] suggests two formulas (which have been slightly adjusted to replace constants with variables) for calculating the power consumption of training deep neural network models for NLP. They estimate kgCO$_2$e/kWh by first measuring the power consumption of different computing components:

$$p_t = \frac{\Omega \cdot t \cdot (p_c + p_r + p_g)}{1000} \tag{1}$$

Where Ω is a coefficient representing the power usage effectiveness (PUE) for where an experiment is run;[1] t is the total running time in hours; p_c is the average power draw from all CPUs utilised; p_r is the average power draw of the memory utilised (i.e., DRAM); and p_g is the average power draw across all GPUs utilised. Note that all the p_x variables are measured in watts, meaning that p is

[1] Strubell et al. use the global average for data centres. However it should be ideally the PUE of the data centre or locality where experiments are run

a measure of kilowatt-hours (kWh). Lannelongue [42] proposes a similar equation to measure power draw, which provides a more accurate estimate; however, we believe that the estimate by Strubell et al. is sufficient and easier to calculate given the existing software libraries that facilitate recording power draw (as there are several limitations with existing libraries as we will show in Section 4.0.1). With this measurement, Strubell et al. then compute the emissions of an experiment as:

$$\text{kgCO}_2\text{e} = \theta \cdot p_t \quad (2)$$

Where θ is the average amount of CO_2 (in kilograms) produced per kilowatt-hour in the region where the experiment took place, this is typically measured as an amount of CO_2 equivalents per unit of power consumed, for example, kgCO$_2$e/kWh. Again, this should ideally be the value for the data centre or locality in which experiments are run, but a global estimate can also be used. Note that it is essential that both Ω, θ, and all other variables used for computing kgCO$_2$e are reported so that others interested in replicating, reproducing, or comparing results can do so. This estimation of carbon emissions can easily be translated to IR experiments (it is not domain-specific), is relatively easy to calculate, and can be used to compare the number of emissions produced by different models, so long as the constants and units are the same.

This measure can also be extended to calculate the CO_2 emissions of a retrieval system in production. Rather than retrospectively measuring how many CO_2 emissions were generated from the training procedure for a given IR model, it can be used to estimate the emissions over the lifetime it may be used in production. In addition to estimating the CO_2 emissions produced training a retrieval model, it is also possible to estimate the CO_2 emissions produced for a single query: p_q. Using the number of queries issued to a search engine over a period of time (e.g., one hour), one can estimate the impact of their retrieval model in production:

$$\text{kgCO}_2\text{e} = \theta \cdot \Delta_q \cdot p_q \quad (3)$$

Where Δ_q is the average number of queries issued to a search engine in one hour.

4.0.1 *Measuring Power Draw.*

Now that we have established how to quantify the carbon emissions for Information Retrieval experiments, we provide suggestions for software libraries that one can use to record the power usage values for computing kgCO$_2$e/kWh. Historically, measuring the energy consumption of hardware and software has been necessary for organisations to manage their energy utilisation. That being said, only recently have libraries been developed to measure the energy impact of scientific experiments. However, even these libraries are limited to the operating system and hardware architectures that are supported. Noureddine et al. [65] provide a survey of energy measurement approaches. They identify three classes of energy measurement approaches: hardware measurements, software measurements, and power models (i.e., estimating the energy usage from hardware and software measurements). They suggested that software measurements (e.g., profilers) were the most promising approach due to the limitations of hardware and power models at the time. However, more recently, energy measurement approaches have evolved to include much more fine-grained measurements of hardware utilisation. For convenience, in Table 1 we have listed several recently published software libraries that can be used to record the necessary variables for computing kgCO$_2$e/kWh. Note that most libraries, including the ones mentioned in Table 1 can only record measurements for Intel CPUs and nVidia GPUs.

4.1 Alternative Measures

In contrast with direct measurements of power usage and emissions, there have also been several alternative measurements that seek to quantify or index how Green research is. Henderson et al. [33] argue that there are also limitations to reporting emissions: emission production can differ depending on the region, time of day, and even across years. Reporting the power consumption allows one to estimate the amount of emissions given the carbon intensity of a specific region or time. Economic measures [18, 43] may also help with understanding costs, but we expect them to correlate with our measures, thus we do not perform this analysis here.

Floating Point Operations (FPO) There are several state-of-the-art methods in the Machine Learning and Natural Language Processing literature that measure computational cost through floating-point operations, or FPO [31, 55, 83, 84]. One advantage of this measure is that the FPO value for any given method will be the same independent of hardware. However, one disadvantage of FPO is that there is no single agreed-upon way to calculate it. Therefore, although each of the papers cited above mentions that they measure the cost of their method through FPOs, the individual methods are not comparable. Compared to measuring power usage, although there can be fluctuations in hardware utilisation, if one uses the same hardware for two different experiments, one can obtain somewhat comparable results (i.e., depending on random factors like time of day, region, and other fluctuations).

Red AI Cost Schwartz et al. [72] propose an equation for estimating the total cost of producing a single result (R): $Cost(R) \propto E \cdot D \cdot H$ where E is the cost of executing the model on a single sample, D is the size of the training dataset; i.e., the number of times the model was executed in training, and H is the number of hyperparameter experiments; i.e., the number of times a model was trained throughout the development of the model. However, like FPO described above, the authors of this measure do not specify precisely the cost of executing a model, i.e., it is left as an exercise to the researcher wanting to use the model. Therefore, the same advantages and disadvantages listed for FPO above exist for this measure. Another disadvantage of this measure is that it primarily captures the training cost of models. For research purposes, this may be satisfactory for reporting in a publication but is lacking for reporting the cost of methods in reality, for example, the cost of retrieval in a search engine over a period of time.

However, the downside to these measures listed above is that they do not translate easily into a meaningful value. Comparing how many emissions are produced by travelling or running household appliances to the emissions produced through research is a much more natural way to understand the impact of the emissions. This is opposed to an index that simply represents the computational cost of research. Here, there is no analogue so as to easily be able to understand the impact of the cost.

5 ENVIRONMENTAL COST OF IR RESEARCH

Using the equations for estimating the emissions produced by IR research, we demonstrate the possible levels of emissions produced by different typical IR research pipelines. We use off-the-shelf libraries and tools publicly available and commonly used by the IR community for all of the experiments. The experiments we conduct all investigate the 'cost' involved in some factors of IR research. By 'cost', we explicitly refer to the *environmental cost* of an experiment, measured as kgCO$_2$e/kWh as defined in Section 4.

Our experiments investigates two main factors: (1) 'traditional' IR research (e.g., inverted index or statistical scoring functions) versus 'neural' IR research (e.g., dense retrievers and neural IR), and (2) offline costs (e.g., indexing documents, training models and fine tuning, hyperparameter tuning) versus online costs (e.g., performing retrieval, scoring documents).

5.1 Experimental Setup

5.1.1 Test Collection. The collection we use to perform these experiments is MS MARCO v1 [61]. Although we acknowledge that using a single collection does not provide a very generalisable result, there are very few collections available with enough topics to support training neural Information Retrieval models, which are the main interest of this paper. MS MARCO is also a highly used collection in the IR community at the moment, and therefore results on this collection will be meaningful to many. One aspect of IR research that we are not considering in our results is the cost associated with network transfers. Distributing large collections such as MS MARCO have an environmental cost associated with them, from storing the collection to facilitating the network transfer of the collection. Unfortunately, we were unable to identify any estimates that quantify this cost. As a result, we cannot make any concrete claims about how these costs may impact the total emissions produced by an experiment. However, we believe that going forward, this cost should be measured and considered by those that use large collections, especially as these collections grow in size and scope into the future: The passage collection of MS MARCO v2 is 15.6 times larger than the original, weighing in at 32.3GB compressed. In our experiments, we use the smaller v1 collection.

5.1.2 Implementation Details. In terms of methods, to reduce the number of comparisons, we investigate a single 'traditional' method: BM25 [70]. Despite using only a single method to represent all 'traditional' IR, BM25 is perhaps the most commonly used retrieval method in IR research, and is often used as an initial ranking for more complex re-rankings [19]. We contrast BM25 (using pyserini [45]) against several IR baselines: dense retrievers (using DPR [39]), sparse retrievers (using uniCOIL [44]), neural re-ranking (TILDEv2 [94] and monoBERT [64]), re-ranking with learning to rank (using LambdaMART [7]), and neural document expansion models (using TILDE [95] and docT5query [63]).

BM25 We use the standard pyserini [45] indexing and retrieval scripts, with b and $k1$ parameters chosen based on prior tuning experiments performed on the MS MARCO collection.

LambdaMART We cannot evaluate this collection using MS MARCO as no learning to rank features exists. Thus, we train a LambdaMART model using the Yahoo! C14B collection, which contains a similar number of training examples as MS MARCO. For evaluation, we create a synthetic collection by oversampling from the C14B test portion to match the number of examples in the dev portion of MS MARCO. We use the implementation from LightGBM [40]. Note that in our calculations we are unable to factor in the cost of feature extraction, which we believe would be considerably expensive, depending on the type and number of features.

DPR We mainly follow the training configuration in the original paper [39] with slightly different parameter setting to train the DPR model. Specifically, we use the `bert-base-uncased` checkpoint offered by Huggingface transformers [87] to build a bi-encoder DPR model and use BM25 hard negative sampling strategy to train the model. We randomly sampled seven hard negative passages from the top 200 passages retrieved by BM25 and one positive passage from the relevance assessments. We set the batch size to 16 and applied in-batch negatives sampling to each training sample in the batch, resulting in $7 + 8 * 15 = 127$ negatives per training sample. We train with the AdamW optimiser and a 5e-6 learning rate and a linear learning rate schedule for 150K updates. For inference, we use the FAISS library [38] to index and retrieve dense vectors.

monoBERT We follow the training practice described by Nogueira and Cho [62]. We fine-tune a `bert-large-uncased` checkpoint from Huggingface transformers with binary cross-entropy loss to perform binary classification on query-passage pairs. Negative pairs are randomly sampled from the top 1,000 passages retrieved by BM25. We set the ratio of positive pairs to negative pairs to 1:4. The model is trained on two Tesla V100 GPUs with a batch size of $2 * 64$ for 70K updates. We use monoBERT to re-rank the top 1,000 passages retrieved by BM25.

TILDEv2 We directly use the training and inference scripts with the same configurations available on the official Github repository [2]. We use TILDEv2 to re-rank the top 1,000 passages from BM25.

uniCOIL We directly use the official training scripts [3] to train the model and use pyserini [4] to index and inference.

Passage expansion For TILDE expansion [95], we directly use the code provided in the official repository [5]. For docTquery expansion, we use the Huggingface transformers implementation that is available in the official repository [6]. Since using docTquery to expand the whole MS MARCO passage collection is very expensive [63, 94], we randomly sampled a subset of the MS MARCO collection with 2,560 passages and only use docTquery and TILDE to expand this subset to estimate the overall running time.

5.1.3 Power and Emission Measurement. We use the HPC cluster available to us at our institution for running each of the experiments. We contacted the manager for the computing infrastructure to obtain the PUE (Ω), which was 1.89. We could not obtain the average emissions produced per hour from this contact, so we used

[2] https://github.com/ielab/TILDE/tree/main/TILDEv2
[3] https://github.com/luyug/COIL/tree/main/uniCOIL
[4] https://github.com/castorini/pyserini/blob/master/docs/experiments-unicoil.md
[5] https://github.com/ielab/TILDE#passage-expansion-with-tilde
[6] https://github.com/castorini/docTTTTTquery#predicting-queries-from-passages-t5-inference-with-pytorch

the $\text{kgCO}_2\text{e/kWh}$ (θ) for our region, 765.9.[7] We record the power usage (in watts) and the running time (in seconds) through the CodeCarbon [71] library and convert the values to their appropriate units before computing Equations 1, 2, and 3 (See Appendix A). Note that in parts of the world with higher adoptions of renewable technologies, the $\text{kgCO}_2\text{e/kWh}$ may be lower than what we report. In any such case, we believe it is still important to report power and emissions as it promotes energy-efficient methods and running experiments on Green infrastructure.

To promote uptake and discourse of Green IR methods, we release all of the code and data used in these calculations for others to reproduce and reuse: https://github.com/ielab/green-ir.

5.2 Results

5.2.1 Impact of Experiments on Emissions. We begin our analysis by first investigating the emissions cost for obtaining a single result. For simplicity, we refer to the concept of an entire Information Retrieval pipeline, from training or tuning; to indexing; to retrieval or re-ranking in order to measure the effectiveness of a method, as an *experiment.* These experiments on the methods described in Section 5.1 are presented in Table 2.

Examining ‘traditional’ pipelines like BM25 and learning to rank, the emissions produced by these methods is several orders of magnitude lower than more recent neural methods. This result suggests that one can perform hundreds of experiments using these methods before reaching the number of emissions produced by ‘modern’ methods. The LTR is so close to BM25 in the number of emissions produced because it is using a linear ranker.

A neural ranker is likely to produce more emissions. In contrast, monoBERT produces ten times more emissions to obtain a single result than other methods when comparing the neural methods. Conversely, DPR produces the least amount of emissions. As discussed below, the amount of emissions produced may be an indirect indication of effectiveness.

Next, we further investigate the emissions produced by each neural method for an experiment. Comparing uniCOIL and TILDEv2, two document expansion models, we found that although the retrieval stage of TILDEv2 is more efficient than uniCOIL, the overall cost of an experiment is higher. Furthermore, these two methods have an expansion stage where we chose to use TILDE and docTquery. The running time of docTquery is almost 70 times that of TILDE, producing over 100 times more emissions. Across all neural methods, excluding the document expansion step, the training step was the most expensive. It produced the most emissions, with monoBERT producing approximately ten times as much during the training process as other neural methods. This is likely because monoBERT is using the `bert-large` model and because it cannot pre-compute document representations, instead needing to estimate relevance at query time. On the other hand, TILDEv2 pre-computes everything at the indexing time, costing less at retrieval time.

To get an understanding of how many emissions are produced by these methods compared to more familiar reference points, Table 3 contains the emissions produced by common household appliances, air travel, and car travel. Obtaining a single result for a given experiment is comparable to the emissions produced through the usage of common household appliances. A single researcher may produce the same emissions as taking a short commercial airline flight throughout model development. A research lab will naturally produce more emissions (although more difficult to estimate).[8]

Experiment	Running Time (hours)	Power (kWh)	Emissions (kgCO_2e)
BM25 Indexing	0.0809	0.0021	0.0016
BM25 Search	0.0025	0.00006	0.00005
	0.0834	**0.0022**	**0.0017**
LambdaMART Training	0.0285	0.0008	0.0006
LambdaMART Rerank + BM25	0.0628	0.0017	0.0013
	0.0914	**0.0024**	**0.0019**
DPR Training	16.46	6.74	5.16
DPR Indexing	2.42	1.04	0.7958
DPR Search	0.4141	0.0002	0.0001
	19.3	**7.78**	**5.96**
monoBERT Training	57.43	57.95	44.38
monoBERT Rerank + BM25	23.18	10.8	8.27
	80.61	**68.75**	**52.65**
TILDEv2 Training	15.73	6.91	5.29
TILDEv2 Indexing	9.44	4.74	3.63
TILDEv2 Rerank + BM25	0.0247	0.0007	0.0005
TILDE Expansion	11.89	1.04	0.7958
	37.08	**12.69**	**9.72**
docTquery Expansion	760.48	169.06	129.49
	785.68	**180.71**	**138.41**
uniCOIL Training	17.97	7.24	5.54
uniCOIL Indexing	3.66	1.95	1.49
uniCOIL Search	0.8966	0.0237	0.0182
TILDE Expansion	11.89	1.04	0.7958
	34.41	**10.25**	**7.85**
docTquery Expansion	760.48	169.06	129.49
	783.01	**178.28**	**136.54**

Table 2: Cost of IR research over the lifetime of a possible experiment. Each stage in the pipeline (i.e., model training, indexing, and searching) is measured separately. The cumulative cost measurements (i.e., running time, power consumption, and emissions) are also shown at the bottom of each set of stages in each pipeline in bold. Both TILDEv2 and uniCOIL have two totals depending on the choice of document expansion method. A more comprehensive breakdown of search cost is visualised in Figure 1.

[7] https://archive.is/quN83, using the values reported for Queensland in January 2022.

[8] Sourcing data for Table 3 was challenging due to limited availability. Appliance data was sourced from ENERGY STAR (https://www.energystar.gov/productfinder/advanced). Specific URLs are listed in our repository. Flight data was sourced from https://www.atmosfair.de/en/offset/flight/. Car data was sourced from https://www.epa.gov/automotive-trends/explore-automotive-trends-data. Accessed in January 2022.

Appliance	Running Time (hours)	Power (kWh)	Emissions (kgCO$_2$e)
Television	5.0000	0.3664	0.2806
Clothes Dryer	1.0847	2.2491	1.7226
Refrigerator	168.0000	7.3544	5.6327
Flight from Frankfurt to Madrid			728
Flight from New York to Madrid			2,293
Flight from Shanghai to Madrid			4,911
Flight from Melbourne to Madrid			11,682
Driving 10,00km by car			5,617

Table 3: Power consumption of common household appliances, for flights between several locations and where SIGIR is hosted this year, and driving 10,000km by car. Note that the power consumption is for a typical consumer household and not a data centre. Therefore, the PUE in Equation 1 is set to 1.0. Note also that the flight emissions are for a *single* passenger. Our code repository contains data sources and scripts used to process it into this format.

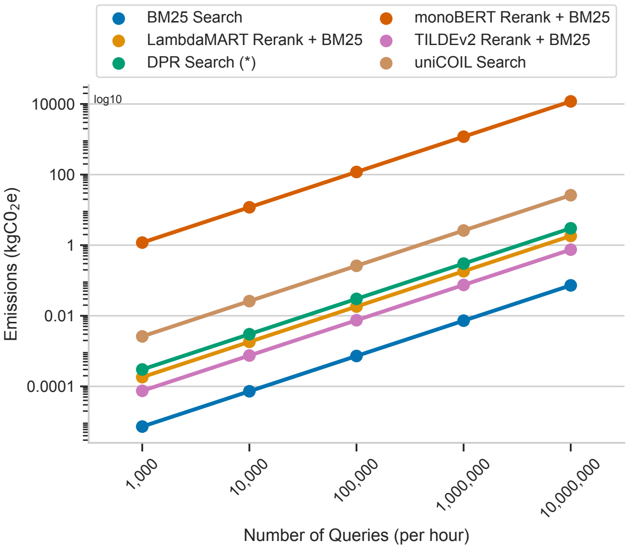

Figure 1: kgCO$_2$e produced per hour for different estimates of queries issued to a hypothetical search engine per hour. Note that the values obtained are estimated from the running time and energy consumption reported in Table 2. Note that DPR has been estimated from running 200 queries instead of the entire 6980 dev set of MS MARCO (query-by-query instead of batch retrieval). Values are highly approximate and do not account for additional factors such as load balancing, caching, or batching.

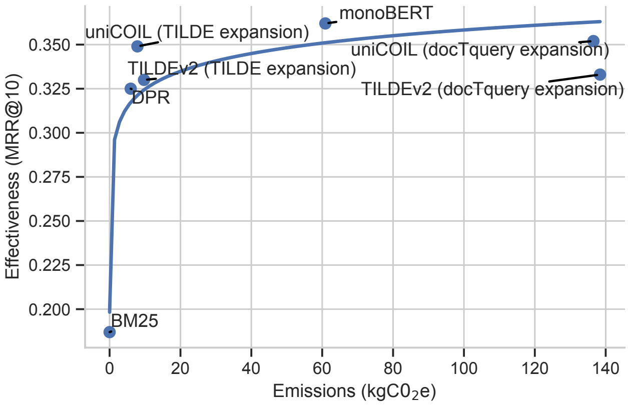

Figure 2: Effectiveness-emissions trade-off for obtaining a single experimental result (i.e., summed rows in Table 2). Note that the learning to rank method is not included in this figure because features were unavailable for MS MARCO documents.

5.2.2 *Cost of Experiment versus Cost in Production.* Research and experiments are often only one side of the coin in IR. The flip side of research is the deployment of IR systems. Using Equation 3, we plot the estimated emissions produced for each of the methods in Table 2, presented in Figure 1. This figure only visualises the search stage and not the entire pipeline. Furthermore, it is unlikely that a single machine will handle tens of millions of queries per hour alone; instead, a distributed system involving many servers is used for load balancing. For some systems such as monoBERT, the estimates are impossible on a single machine, as evidenced in Table 2 where only approximately 300 queries can be processed per hour (dev size/running time). Therefore the estimates offer a perspective on a likely lower bound in the emissions produced that do not consider additional resources that would be used to load-balance a production system. Additionally, the x-axis in Figure 1 only goes to 10,000,000 queries per hour. This is a low value compared to the traffic popular search engines may receive, although finding a reliable number is difficult, so our estimates are conservative.[9]

Given our estimates for the number of emissions produced by a given search system, we found that even at large scales on the order of millions of queries per hour, most methods produce relatively low amounts of emissions. With our approximate, lower-bound estimates, the second worst method in terms of emissions, uniCOIL, could be run in a production system for an entire year and not produce the same amount of emissions as a flight from Melbourne to Madrid; monoBERT, on the other hand, would produce approximately the same amount in a single hour. Finally, although TILDEv2 produces more emissions to achieve a single experimental result (as in Table 2), in production, it is the second best behind BM25. This finding suggests that there can be a trade-off also in producing more emissions to train a model where it will produce lower emissions in production over the lifetime of the model. Note that in production systems, it is likely that query results would be cached to avoid needing to perform an end-to-end query processing pipeline. We leave such investigations for future work.

[9] DuckDuckGo shares official traffic statistics (https://duckduckgo.com/traffic), however they have a relatively low market share.

5.2.3 Effectiveness-Emissions Trade-off. As alluded to earlier, we observe a trade-off in the number of emissions produced for an experiment and the effectiveness that the experiment obtains. Figure 2 highlights the trend that we observe for each of the methods listed in Table 2 (excluding the learning to rank experiment given the explanation in Section 5.1.2). This figure suggests that in order to achieve more effective results, more power must be used, and thus more emissions may be produced. This is particularly evident when comparing the expansion methods of TILDEv2 and uniCOIL. The docTquery expansion method results in a higher MRR@10 with a considerable trade-off in emissions.

Notably, the amount of emissions produced appears to plateau as effectiveness increases (as indicated by the blue trend line). This trend suggests that large improvements in effectiveness can be achieved when using neural methods compared to traditional methods such as BM25. However, minor improvements in effectiveness between such models come at the expense of relatively high power usage and thus emission production.

6 DISCUSSION AND LIMITATIONS

Our experiments have demonstrated that compared to other research domains, IR produces relatively low emissions. Even the most demanding methods we considered such as document expansion using docTquery do not produce levels of emissions similar to the air travel for a single passenger. However, one thing to note is that there is a cost associated with the training of the underlying neural models. Results reported by Strubell et al. [77] demonstrate that (pre-)training large language models can be a very costly exercise. Currently, some of the most effective methods on the MS MARCO leaderboard in fact do some amount of pre-training for the underlying language model, e.g., coCondenser [28]. When taking into account extensive model pre-training, we expect to obtain levels of emissions similar to those reported by Strubell et al..

We also note a number of limitations of our study. To begin, we only included a single 'traditional' model (BM25), when we could have also included several, such as QLM [67] or SDM [54]. We believe that BM25 is representative of many of these traditional methods both in terms of effectiveness and emissions. Furthermore, BM25 is also almost exclusively used as an initial ranking for many methods using the MS MARCO collection.

Another set of models that we did not investigate include word embedding-based methods, such as GLM [26] or NTLM [96]. There are also older neural models that exploit convolutional neural networks (CNN) or recurrent neural networks (RNN) [32, 36, 60, 73–76, 90, 91]. We did not use these models because they are smaller than current models (in terms of the number of parameters), and our focus for this perspective paper is on current trends in IR. Compared to current methods, these older neural models are both faster to train and score documents, yet produce results closer to BM25 than to current methods [34] like DPR, TILDEv2, or uniCOIL.

Other limitations include: (1) our focus on passage retrieval, not document retrieval, where neural methods may be more computationally expensive depending on how passage aggregation or document representation is handled, and; (2) our focus on the ad hoc search task, which may be less computationally expensive than other tasks, e.g., in the diversity task, many methods involve comparisons between documents that have already been ranked.

7 CONSIDERATIONS FOR GREEN IR

Although we have suggested that the holistic LCA approach to measuring the emissions produced by IR experiments is infeasible, we believe that one should still consider the emissions produced through the life cycle of an IR experiment. Furthermore, while we also believe that the power usage of the experiments we have demonstrated constitute the primary sources of emissions from IR research, there may still be non-negligible costs that can be attributed to other factors such as data storage. Rather than attempting to measure all of these various factors, we instead provide a framework for IR practitioners to remain mindful of the potential costs. Our framework is inspired by 'reduce, reuse, recycle' campaigns often used for waste management and environmental sustainability. We could not find any reference to this framework being applied to similar domains. However, we believe these three concepts can encapsulate many aspects of Green IR systems. We frame these three concepts (reduce, reuse, and recycle) as a way of pursuing Green IR research. Within each of the subsections below, we also include salient examples from the literature that demonstrate approaches to tackle the challenge each concept represents and possible directions for future work that go deeper into the respective concept. The methods and examples presented in this section are not explicitly Green methods, which have already been listed in Section 3.2. To this end, the examples in this section are intended to be representative of the reduce, reuse, and recycle concepts. We have selected these examples to be the most intuitive from a larger pool. Therefore the reader is urged to use them merely as a starting point for reflecting on their own techniques or methods.

First, however, we provide some analogies to give an intuitive sense of what exactly is meant when referring to reduce, reuse, and recycle concepts. Rather than illustrate these concepts with computer science terminology, we paint a picture using a jam jar. Imagine looking at the shelves in the local jam shop. Of course, there are many flavours of jam to choose from and many sizes of jars. Being both an avid jam lover and mindful of the resources used to create and dispose of jars, thus wanting to **reduce** waste, you choose a larger jar over a smaller jar, so you need to buy new jars less often. After a month, all of the jam in the jar has been used up, and it is time to buy more jam. Rather than buying a new jar, you **reuse** the large jar you bought last time and fill it with new jam from the jam shop. One day, you suddenly realise that you have lost your love of jam. Your new passion in life is candle making, and you **recycle** your trusty jam jar, filling it with candle wax.

To use a more concise description of these concepts within the context of IR research: to **reduce** is to expend fewer resources, to **reuse** is to repurpose resources intended for one task to the same task, and to **recycle** is to repurpose resources intended for one task to a different task. Finally, although our objective with these concepts is to provide a guiding framework for considering the impact of emissions on IR research, they also promote a more supportive research environment. For example, publicising failed research reduces duplicated work and sharing a pre-trained model for reuse ensures reproducible and replicable research.

7.1 Reduce

The first consideration that can be made when designing experiments in a Green way is often the most straightforward: simply reduce the number of experiments involved. However, as the examples demonstrate, the concept of reduction can also be thought of as limiting expensive computations, e.g., utilising the CPU over GPU. To this end, one can think about this concept not in terms of experimentation but in terms of available *resources*. As such, prior to starting any research or experiments, one may ask themself: *How can I perform research with fewer resources?*

7.1.1 Random Hyper-parameter Search. Many experiments involving hyper-parameters require optimisation to identify the most effective model. For the majority of experiments that involve hyper-parameter optimisation, it has been empirically and theoretically shown that random search is computationally more efficient than grid search [5]. Furthermore, reducing the search size naturally reduces the number of resources required for experimentation.

7.1.2 CPU-based Inference. Typical transformer-based deep learning ranking models require specialised GPU hardware for efficient query processing. The utilisation of the GPU for this purpose is considerably more energy-intensive than using the CPU [64]. To address this problem, several papers have spawned to attempt to repurpose existing models to permit CPU-based inference for the effective and efficient ranking of documents. One key example of this is docT5query [63] which uses a fine-tuned T5 model for expanding documents prior to indexing them for retrieval by traditional models such as BM25. A more recent example of this approach is TILDEv2 [94] which combines the offline document expansion step with a fast CPU-based query likelihood scoring mechanism.

7.2 Reuse

The second consideration that can be made when designing experiments in a Green way is to reuse existing software artefacts such as data, code, or models. Reusing existing technology and data means taking something existing and repurposing it for a task not initially intended. As such, prior to starting any research or experiments, one may ask themself: *How can I repurpose data, code, or other digital artefacts meant for one task to the same task?*

7.2.1 Reuse Large Collections. Rather than recreating entirely new collections for each task, a common approach to developing new methods or tasks is to reuse existing collections. One prominent example of this in the Information Retrieval community is the MS MARCO collection which was initially intended for developing and evaluating Natural Language Processing methods such as question answering, summarisation, and reading comprehension. The TREC Deep Learning track [19, 20] exploits the underlying collection of passages for ad-hoc retrieval research but utilises a different set of queries and deeper relevance assessments. Reusing data in this way reduces the energy required to build the collection in the first place (e.g., scraping web pages or data processing). However, it may also reduce the number of network transfers. The collection was already popular before introducing the TREC Deep Learning track, meaning that many research groups likely already had the majority of the data downloaded.

7.2.2 Pre-indexing Common Collections. As the indexing results have shown in Section 5.2, this step in the IR pipeline, even for traditional inverted indexes, can contribute to measurable amounts of emissions. Reusing indexes reduces emissions produced by indexing the same collection many times (i.e., by researchers, practitioners, and students). It promotes reproducibility by fixing this crucial step in the IR pipeline (although doing so may reduce the diversity of systems when pooling). Both pyterrier [51] and pyserini [45] provide several pre-indexed collections available to be simply downloaded. Further, pyterrier pipelines can be integrated with pyserini indexes. Lin et al. [46] have also proposed a common index file format (CIFF) which promotes interoperability between different search engine implementations [50, 53, 56]. The CLEF eHealth consumer health search workshops [29, 30, 66, 97] are another example of sharing pre-indexed collections for participants to the workshops. In addition to utilising ClueWeb collections, participants in these workshops were able to download a pre-indexed version of the collection, which reduces the amount of time and energy by a factor of how many participants chose this option.

7.3 Recycle

Finally, the last consideration that can be made when designing experiments in a Green way is to recycle existing software artefacts such as data, code, and models. Recycling existing technology and data is subtly different from reusing such artefacts. We differentiate reuse from recycle by defining recycle as the action of *repurposing* an existing artefact for a task it was not originally intended. The repurposing of artefacts does not necessitate the modification typically required when, for example, reusing a model. As such, prior to starting any research or experimentation, one may ask themself: *How can I repurpose existing data, code, or other digital artefacts meant for one task to a different task?*

7.3.1 Neural Query Expansion. Rather than fine-tuning language models for a particular Information Retrieval task, there have also been some methods that have exploited the pre-trained language model directly. This recycling of a model, therefore, requires no pre-training step. In the IR domain, there have been at least two such studies that utilise pre-trained transformer models for query expansion [16, 58]. The intuition of these methods is to exploit the textual generation ability of neural language models to add relevant terms to a query. Although these methods avoid expensive pre-training and fine-tuning neural language models, they still may be unsuitable for production settings, requiring specialised GPU hardware for efficient query processing. There is a clear direction for future work that exploits these pre-trained models for effective and power-efficient expansion.

7.3.2 Passage expansion with TILDE. Another example of the recycle concept is the TILDE model for passage expansion. TILDE is a BERT-based language model that is trained with query likelihood. It can predict relevant query tokens given a passage. Originally, TILDE was used to re-rank passages [95], however the authors found that it can also serve as a passage expansion model just like docTquery [94] without further fine-tuning. As we demonstrated in Section 5.2, TILDE can be used for ranking models that rely on passage expansion and has a similar retrieval effectiveness as docTquery while producing fewer emissions.

8 CONCLUSION

To summarise the contributions of this perspective paper, (1) we have provided a literature review of Green methods in not only the IR domain but in closely related domains such as NLP and ML, where we found a lack of research into quantifying the emissions for IR, (2) we have provided a framework for IR practitioners to consider when designing experiments and systems by contextualising the 'reduce, reuse, recycle' concepts to the IR domain, and (3) we have undertaken an investigation into the power usage and emissions produced by several well-known IR ranking pipelines. We have found that the current trends in IR do not lead to excessive emissions and likely lower emissions than those produced by research in related domains like NLP. Further, we note that models requiring more resources achieve increasingly smaller gains in retrieval effectiveness.

Current Costs and Impact of IR Research. We have focused our attention on what we believe to be the most resource-intensive (and thus the most emissive) aspect of IR. However, the ranking models that have been investigated in this paper are not the only contributor to emissions. There is still much research to be done in increasing the efficiency of existing traditional search systems for massive scalability and in other areas such as distributed search [80]. We also recognise that the field of IR is not limited to ad hoc retrieval in the context of the web, where the experiments of this paper have been focused. However, for these related tasks and new ranking models, our measures for quantifying emissions, which are based on the work of Strubell et al. [77] and our conceptual framework for making considerations for Green IR, can be used to assess and compare their impact.

Implications for Future Directions in IR Research. The continued uptake of larger and larger neural methods that require specialised and power-hungry hardware and increasing amounts of data will drive the demand for the utilisation of more power and more data. Further to this point, we believe that given the current direction of IR research, several orders of magnitude more power usage and data may be required in the future (and thus more emissions). One clear example of this effect (outside of the results we have shared in this paper) is the search results from a recent OpenAI paper [59]. Their largest model with 175 billion parameters fails to outperform relatively old neural models that have on the order of 100 million parameters on several collections from the BEIR evaluation suite [79]. It is also unclear from the paper how long it takes to train and make predictions with these larger models, likely requiring specialised hardware for both. From this method, we posit that the current trend of increasing the size of models to achieve higher effectiveness may not apply to IR research as it does in the NLP and ML domains.

We believe that a natural future direction in IR research is to go beyond the current trend of fine-tuning large pre-trained language models initially designed for NLP tasks. Techniques such as Condenser [27] and PROP [48] highlight that pre-trained language models such as BERT may not be ideal for fine-tuning ranking models. Indeed, compared to methods fine-tuned on BERT, these methods propose pre-training tasks that result in language models that produce highly effective rankers once fine-tuned. From these methods, we believe that one clear direction for future research in the IR domain is to develop more effective pre-trained models specialised for an IR task that can be fine-tuned with less data to other tasks or domains. Training these new IR-focused models may require similar amounts of data and computation to those seen in the NLP domain. Thus requiring more power and possibly producing emissions in line with what is generated when pre-training models like BERT.

Lastly, at the time of writing, there have arisen two end-to-end transformer models that encapsulate the entire indexing and searching architecture into a single model [78, 93]. These techniques represent a paradigm shift in how retrieval systems are developed. No longer does a separate index need to be maintained. Instead, a single model can be deployed to retrieve and rank documents. These methods demonstrate that a single end-to-end model can replace the traditional 'retrieve and rank' paradigm of ad hoc search. However, two main aspects restrict the feasibility of such models: the model's size is likely to grow with the number of documents in the corpus (the two works mentioned above use relatively small collections), and the model's effectiveness depends on the number of parameters (in particular, DSI [78] requires hundreds of highly specialised TPUv4 units). These models also present a new set of challenges, such as how to scale these models to large collections, handle adding new documents without having to reindex the entire collection (and thus re-train the model), efficiently index documents, and process large numbers of queries.

Summary. As a community, we must be mindful of the potential costs that our research may have. The ways that we measure and address the environmental impact of our research are just one of the many brushstrokes that coalesce into a larger landscape that portray our impacts on society at large. Our emission quantification measures for assessing the impact of obtaining an experimental result and our framework for making considerations about the Green-ness of an IR system can be used to make decisions about and devise experiments for future IR research and practice.

The IR community is at a turning point in terms of the types of deep learning models used, the scale of those models, and how those models are trained. While the investigation into and development of such models are valuable research goals, we believe that it is important to be mindful of the costs and environmental impacts of these techniques. As the development of new IR-focused deep learning models grows, similar trends in terms of costs and environmental impact that have been seen in other research communities may appear. Given the trend in highly sophisticated neural models for search, we believe that Green IR will become an increasingly important aspect of IR research.

We firmly believe that the tools and framework presented in this paper provide a solid foundation that others may use to understand the impact of their experimentation and to reason about devising and considering Green Information Retrieval research.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their insightful feedback in further shaping the paper. We would also like to thank Dr Joel Mackenzie, Dr Ahmed Mourad, and Dr Daniel Locke for their helpful comments on drafts of this paper.

REFERENCES

[1] ISO/TC 207/SC 5. 2006. *ISO 14040:2006 Environmental Management — Life Cycle Assessment — Principles and Framework* (second ed.). ISO, Geneva, Switzerland.

[2] Susanne Albers. 2010. Energy-Efficient Algorithms. *Commun. ACM* 53, 5 (2010), 86–96.

[3] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. 2020. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models. *arXiv preprint arXiv:2007.03051* (July 2020). arXiv:2007.03051

[4] Lotfi Belkhir and Ahmed Elmeligi. 2018. Assessing ICT Global Emissions Footprint: Trends to 2040 & Recommendations. *Journal of Cleaner Production* 177 (March 2018), 448–463.

[5] James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-Parameter Optimization. *Journal of machine learning research* 13, 2 (2012).

[6] Roi Blanco, Matteo Catena, and Nicola Tonellotto. 2016. Exploiting Green Energy to Reduce the Operational Costs of Multi-Center Web Search Engines. In *Proceedings of the 25th International Conference on World Wide Web.* 1237–1247.

[7] Christopher JC Burges. 2010. From Ranknet to Lambdarank to Lambdamart: An Overview. *Learning* 11, 23-581 (2010), 81.

[8] Matteo Catena. 2015. Energy Efficiency in Web Search Engines. In *Sixth BCS-IRSG Symposium on Future Directions in Information Access (FDIA 2015) 6.* 1–2.

[9] Matteo Catena, Ophir Frieder, and Nicola Tonellotto. 2018. Efficient Energy Management in Distributed Web Search. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* 1555–1558.

[10] Matteo Catena, Craig Macdonald, and Nicola Tonellotto. 2015. Load-Sensitive CPU Power Management for Web Search Engines. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 751–754.

[11] Matteo Catena and Nicola Tonellotto. 2015. A Study on Query Energy Consumption in Web Search Engines.. In *Proceedings of the 6th Italian Information Retrieval Workshop.*

[12] Matteo Catena and Nicola Tonellotto. 2017. Energy-Efficient Query Processing in Web Search Engines. *IEEE Transactions on Knowledge and Data Engineering* 29, 7 (2017), 1412–1425.

[13] Gobinda Chowdhury. 2012. An Agenda for Green Information Retrieval Research. *Information Processing & Management* 48, 6 (Nov. 2012), 1067–1077.

[14] Gobinda Chowdhury. 2013. Sustainability of Digital Information Services. *Journal of Documentation* 69, 5 (Jan. 2013), 602–622.

[15] Gobinda Chowdhury. 2014. Sustainability of Digital Libraries: A Conceptual Model and a Research Framework. *International Journal on Digital Libraries* 14, 3 (Aug. 2014), 181–195.

[16] Vincent Claveau. 2021. Neural Text Generation for Query Expansion in Information Retrieval. *WI-IAT 2021 - 20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (Dec. 2021), 1–8.

[17] Matt Crane, J. Shane Culpepper, Jimmy Lin, Joel Mackenzie, and Andrew Trotman. 2017. A Comparison of Document-at-a-Time and Score-at-a-Time Query Evaluation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.* ACM, Cambridge United Kingdom, 201–210.

[18] Nick Craswell, Francis Crimmins, David Hawking, and Alistair Moffat. 2004. Performance and Cost Tradeoffs in Web Search. In *Proceedings of the 15th Australasian Database Conference-Volume 27.* 161–169.

[19] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the Trec 2020 Deep Learning Track. *arXiv preprint arXiv:2003.07820* (2020). arXiv:2003.07820

[20] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2019. Overview of the Trec 2019 Deep Learning Track. *arXiv preprint arXiv:2102.07662* (2019). arXiv:2102.07662

[21] Zhuyun Dai and Jamie Callan. 2019. Context-Aware Sentence/Passage Term Importance Estimation for First Stage Retrieval. *arXiv preprint arXiv:1910.10687* (2019). arXiv:1910.10687

[22] 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, Volume 1 (Long and Short Papers).* Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186.

[23] Göran Finnveden, Michael Z. Hauschild, Tomas Ekvall, Jeroen Guinée, Reinout Heijungs, Stefanie Hellweg, Annette Koehler, David Pennington, and Sangwon Suh. 2009. Recent Developments in Life Cycle Assessment. *Journal of environmental management* 91, 1 (2009), 1–21.

[24] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, Virtual Event Canada, 2288–2292.

[25] Ana Freire, Craig Macdonald, Nicola Tonellotto, Iadh Ounis, and Fidel Cacheda. 2014. A Self-Adapting Latency/Power Tradeoff Model for Replicated Search Engines. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining.* 13–22.

[26] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth JF Jones. 2015. Word Embedding Based Generalized Language Model for Information Retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 795–798.

[27] Luyu Gao and Jamie Callan. 2021. Condenser: A Pre-training Architecture for Dense Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 981–993.

[28] Luyu Gao and Jamie Callan. 2021. Unsupervised Corpus Aware Language Model Pre-Training for Dense Passage Retrieval. *arXiv preprint arXiv:2108.05540* (2021). arXiv:2108.05540

[29] Lorraine Goeuriot, Gareth Jones, Liadh Kelly, Johannes Leveling, Allan Hanbury, Henning Muller, Sanna Salantera, Hanna Suominen, and Guido Zuccon. 2013. ShARe/CLEF eHealth Evaluation Lab 2013, Task 3: Information Retrieval to Address Patients' Questions When Reading Clinical Reports. In *Proceedings of the CLEF 2013 Workshop on Cross-Language Evaluation of Methods, Applications, and Resources for eHealth Document Analysis*, D. Tufis, P. Forner, and R. Navagli (Eds.). The CLEF Initiative (Conference and Labs of the Evaluation Forum), Spain, 1–16.

[30] Lorraine Goeuriot, Liadh Kelly, Wei Li, Joao Palotti, Pavel Pecina, Guido Zuccon, Allan Hanbury, Gareth J. F. Jones, and Henning Müller. 2014. ShARe/CLEF eHealth Evaluation Lab 2014, Task 3: User-centred Health Information Retrieval. In *Proceedings of CLEF 2014.* Sheffield, United Kingdom.

[31] Ariel Gordon, Elad Eban, Ofir Nachum, Bo Chen, Hao Wu, Tien-Ju Yang, and Edward Choi. 2018. Morphnet: Fast & Simple Resource-Constrained Structure Learning of Deep Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1586–1595.

[32] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-Hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* 55–64.

[33] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. *Journal of Machine Learning Research* 21, 248 (2020), 1–43.

[34] Sebastian Hofstätter and Allan Hanbury. 2019. Let's Measure Run Time! Extending the IR Replicability Infrastructure to Include Performance Aspects. *arXiv preprint arXiv:1907.04614* (2019). arXiv:1907.04614

[35] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2020. Interpretable & Time-Budget-Constrained Contextualization for Re-Ranking. In *24th European Conference on Artificial Intelligence (Frontiers in Artificial Intelligence and Applications, Vol. 325).* IOS Press, Santiago de Compostela, Spain.

[36] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management.* 2333–2338.

[37] Paras Jain, Xiangxi Mo, Ajay Jain, Alexey Tumanov, Joseph E. Gonzalez, and Ion Stoica. 2019. The OoO VLIW JIT Compiler for GPU Inference. *arXiv preprint arXiv:1901.10008* (Jan. 2019). arXiv:1901.10008

[38] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-Scale Similarity Search with Gpus. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[39] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, Online, 6769–6781.

[40] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A Highly Efficient Gradient Boosting Decision Tree. *Advances in neural information processing systems* 30 (2017).

[41] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the Carbon Emissions of Machine Learning. *arXiv preprint arXiv:1910.09700* (2019). arXiv:1910.09700

[42] Loïc Lannelongue, Jason Grealey, and Michael Inouye. 2021. Green Algorithms: Quantifying the Carbon Footprint of Computation. *Advanced Science* 8, 12 (2021), 2100707.

[43] Kewen Liao, Alistair Moffat, Matthias Petri, and Anthony Wirth. 2017. A Cost Model for Long-Term Compressed Data Retention. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.* 241–249.

[44] Jimmy Lin and Xueguang Ma. 2021. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. *arXiv preprint arXiv:2106.14807* (2021). arXiv:2106.14807

[45] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, Virtual Event Canada, 2356–2362.

[46] Jimmy Lin, Joel Mackenzie, Chris Kamphuis, Craig Macdonald, Antonio Mallia, Michał Siedlaczek, Andrew Trotman, and Arjen de Vries. 2020. Supporting Interoperability Between Open-Source Search Engines with the Common Index File Format. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, Virtual Event China, 2149–2152.

[47] Kadan Lottick, Silvia Susai, Sorelle A. Friedler, and Jonathan P. Wilson. 2019. Energy Usage Reports: Environmental Awareness as Part of Algorithmic Accountability. In *Workshop on Tackling Climate Change with Machine Learning at the 33rd Conference on Neural Information Processing Systems*. Vancouver, Canada. arXiv:1911.08354

[48] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining.* ACM, Virtual Event Israel, 283–291.

[49] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via Prediction of Importance with Contextualization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1573–1576.

[50] Craig Macdonald, Richard McCreadie, Rodrygo LT Santos, and Iadh Ounis. 2012. From Puppy to Maturity: Experiences in Developing Terrier.. In *OSIR@ SIGIR.* 60–63.

[51] Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. 2021. PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* ACM, Virtual Event Queensland Australia, 4526–4533.

[52] Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning Passage Impacts for Inverted Indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1723–1727.

[53] Antonio Mallia, Michal Siedlaczek, and Torsten Suel. 2019. PISA: Performant Indexes and Search for Academia. In *Proceedings of the Open-Source IR Replicability Challenge*, Joel Mackenzie (Ed.). 7.

[54] Donald Metzler and W. Bruce Croft. 2005. A Markov Random Field Model for Term Dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 472–479.

[55] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016. Pruning Convolutional Neural Networks for Resource Efficient Inference. *arXiv preprint arXiv:1611.06440* (2016). arXiv:1611.06440

[56] Hannes Mühleisen, Thaer Samar, Jimmy Lin, and Arjen de Vries. 2014. Old Dogs Are Great at New Tricks: Column Stores for Ir Prototyping. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '14).* Association for Computing Machinery, New York, NY, USA, 863–866.

[57] Rakshit Naidu, Harshita Diddee, Ajinkya Mulay, Aleti Vardhan, Krithika Ramesh, and Ahmed Zamzam. 2021. Towards Quantifying the Carbon Emissions of Differentially Private Machine Learning. *arXiv preprint arXiv:2107.06946* (July 2021). arXiv:2107.06946

[58] Shahrzad Naseri, Jeffrey Dalton, Andrew Yates, and James Allan. 2021. CEQE: Contextualized Embeddings for Query Expansion. In *Advances in Information Retrieval (Lecture Notes in Computer Science)*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer International Publishing, Cham, 467–482.

[59] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and Code Embeddings by Contrastive Pre-Training. *arXiv preprint arXiv:2201.10005* (Jan. 2022). arXiv:2201.10005

[60] Gia-Hung Nguyen, Lynda Tamine, Laure Soulier, and Nathalie Bricon-Souf. 2016. Toward a Deep Neural Approach for Knowledge-Based Ir. *arXiv preprint arXiv:1606.07211* (2016). arXiv:1606.07211

[61] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. In *CoCo@ NIPS.*

[62] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019). arXiv:1901.04085

[63] Rodrigo Nogueira and Jimmy Lin. 2019. From Doc2query to docTTTTTquery. (2019), 3.

[64] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with Bert. *arXiv preprint arXiv:1910.14424* (2019). arXiv:1910.14424

[65] Adel Noureddine, Romain Rouvoy, and Lionel Seinturier. 2013. A Review of Energy Measurement Approaches. *ACM SIGOPS Operating Systems Review* 47, 3 (Nov. 2013), 42–49.

[66] João RM Palotti, Guido Zuccon, Lorraine Goeuriot, Liadh Kelly, Allan Hanbury, Gareth JF Jones, Mihai Lupu, and Pavel Pecina. 2015. CLEF eHealth Evaluation Lab 2015, Task 2: Retrieving Information About Medical Symptoms.. In *CLEF (Working Notes).* 1–22.

[67] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 275–281.

[68] Xinchi Qiu, Titouan Parcollet, Daniel Beutel, Taner Topal, Akhil Mathur, and Nicholas Lane. 2020. Can Federated Learning Save the Planet?. In *NeurIPS-Tackling Climate Change with Machine Learning.*

[69] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* 5835–5847.

[70] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *NIST Special Publication* (1995).

[71] Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni. 2021. CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing. (2021).

[72] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM* 63, 12 (Nov. 2020), 54–63.

[73] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 373–382.

[74] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management.* 101–110.

[75] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning Semantic Representations Using Convolutional Neural Networks for Web Search. In *Proceedings of the 23rd International Conference on World Wide Web.* 373–374.

[76] Alessandro Sordoni, Yoshua Bengio, and Jian-Yun Nie. 2014. Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[77] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, Florence, Italy, 3645–3650.

[78] Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer Memory as a Differentiable Search Index. *arXiv preprint arXiv:2202.06991* (Feb. 2022). arXiv:2202.06991

[79] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).*

[80] Nicola Tonellotto, Craig Macdonald, and Iadh Ounis. 2018. Efficient Query Processing for Scalable Web Search. *Foundations and Trends® in Information Retrieval* 12, 4-5 (2018), 319–500.

[81] Tristan Trébaol. 2020. *CUMULATOR—a Tool to Quantify and Report the Carbon Footprint of Machine Learning Computations and Communication in Academia and Healthcare.* Technical Report.

[82] Andrew Trotman. 2003. Compressing Inverted Files. *Information Retrieval* 6, 1 (2003), 5–19.

[83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems.* 5998–6008.

[84] Tom Veniat and Ludovic Denoyer. 2018. Learning Time/Memory-Efficient Deep Architectures with Budgeted Super Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3492–3500.

[85] Philipp Wiesner, Ilja Behnke, Dominik Scheinert, Kordian Gontarska, and Lauritz Thamsen. 2021. Let's Wait Awhile: How Temporal Workload Shifting Can Reduce Carbon Emissions in the Cloud. In *Proceedings of the 22nd International Middleware Conference.* ACM, Québec city Canada, 260–272.

[86] I.H. Witten, A. Moffat, and T.C. Bell. 1995. Managing Gigabytes: Compressing and Indexing Documents and Images. *IEEE Transactions on Information Theory* 41, 6 (Nov. 1995), 2101.

[87] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45.

[88] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.

[89] Jingjing Xu, Wangchunshu Zhou, Zhiyi Fu, Hao Zhou, and Lei Li. 2021. A Survey on Green Deep Learning. *arXiv preprint arXiv:2111.05193* (Nov. 2021). arXiv:2111.05193

[90] Ju Yang, Jiancong Tong, Rebecca J. Stones, Zhaohua Zhang, Benjun Ye, Gang Wang, and Xiaoguang Liu. 2016. Selective Term Proximity Scoring via Bp-Ann. *arXiv preprint arXiv:1606.07188* (2016). arXiv:1606.07188

[91] Xugang Ye, Zijie Qi, and Dan Massey. 2015. Learning Relevance from Click Data via Neural Network Based Similarity Models. In *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 801–806.

[92] Mirza Yusuf, Praatibh Surana, Gauri Gupta, and Krithika Ramesh. 2021. Curb Your Carbon Emissions: Benchmarking Carbon Emissions in Machine Translation. *arXiv preprint arXiv:2109.12584* (Oct. 2021). arXiv:2109.12584

[93] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, and Ji-Rong Wen. 2022. DynamicRetriever: A Pre-training Model-based IR System with Neither Sparse nor Dense Index. *arXiv preprint arXiv:2203.00537* (March 2022). arXiv:2203.00537

[94] Shengyao Zhuang and Guido Zuccon. 2021. Fast Passage Re-ranking with Contextualized Exact Term Matching and Efficient Passage Expansion. *arXiv preprint arXiv:2108.08513* (Sept. 2021). arXiv:2108.08513

[95] Shengyao Zhuang and Guido Zuccon. 2021. TILDE: Term Independent Likelihood moDEl for Passage Re-ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1483–1492.

[96] Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. 2015. Integrating and Evaluating Neural Word Embeddings in Information Retrieval. In *Proceedings of the 20th Australasian Document Computing Symposium*. 1–8.

[97] Guido Zuccon, Joao Palotti, Lorraine Goeuriot, Liadh Kelly, Mihai Lupu, Pavel Pecina, Henning Mueller, Julie Budaher, and Anthony Deacon. 2016. The IR Task at the CLEF eHealth Evaluation Lab 2016: User-Centred Health Information Retrieval. (2016).

A POWER AND CARBON TRACKING

Although we list several libraries for calculating the power usage (and subsequently the $kgCO_2e$) in Table 1, we exclusively use CodeCarbon [71] for our experiments. Below (Figures 3, 4, and 5) are listed the three different ways that the library can be used to measure power usage information in experiments. The library produces a csv file that records the power usage, estimated carbon emissions, and other information such as when the experiment was run and the duration. From the output, we only use the power usage and duration and calculate $kgCO_2e$ using Equation 1. We provide these code snippets to facilitate the tracking of power usage and estimated carbon emissions in retrieval experiments. Interested researchers can also inspect the code of our experiments where we used this library that we have made available at https://github.com/ielab/green-ir.

```
from codecarbon import track_emissions

@track_emissions()
def experiment()
        # Experiment code goes here
```

Figure 3: Using a decorator to measure a function.

```
from codecarbon import EmissionsTracker

tracker = EmissionsTracker()
tracker.start()
# Experiment code goes here
tracker.stop()
```

Figure 4: Using inline functions to measure arbitrary code.

```
from codecarbon import EmissionsTracker

with EmissionsTracker() as tracker:
        # Experiment code goes here
```

Figure 5: Using a context manager to measure arbitrary code.