Evaluating the carbon footprint of NLP methods: a survey and analysis of existing tools

Nesrine Bannour, Sahar Ghannay, Aurélie Névéol, Anne-Laure Ligozat

Evaluating the carbon footprint of NLP methods: a survey and analysis of existing tools

Nesrine Bannour
Sahar Ghannay
Aurélie Névéol
Université Paris-Saclay, CNRS,
LISN, Orsay, France

Anne-Laure Ligozat
Université Paris-Saclay, CNRS,
ENSIIE, LISN, Orsay, France
firstname.lastname
@lisn.upsaclay.fr

Abstract

Modern Natural Language Processing (NLP) makes intensive use of deep learning methods because of the accuracy they offer for a variety of applications. Due to the significant environmental impact of deep learning, cost-benefit analysis including carbon footprint as well as accuracy measures has been suggested to better document the use of NLP methods for research or deployment. In this paper, we review the tools that are available to measure energy use and CO_2 emissions of NLP methods. We describe the scope of the measures provided and compare the use of six tools (carbon tracker, experiment impact tracker, green algorithms, ML CO2 impact, energy usage and cumulator) on named entity recognition experiments performed on different computational set-ups (local server vs. computing facility). Based on these findings, we propose actionable recommendations to accurately measure the environmental impact of NLP experiments.

1 Introduction

Modern Natural Language Processing (NLP) makes intensive use of deep learning methods because of the functional performance they offer for a variety of tasks, including text classification or named entity recognition (Tourille et al., 2018).

Deep learning programs can have a high environmental impact in terms of Greenhouse Gas (GHG) emissions due in particular to the energy consumption of the computational facilities used to run them (Strubell et al., 2019). The impact has been increasing over the years (Schwartz et al., 2019) and is affecting populations that can be different from those generating the impact (Bender et al., 2021). In a recent medical imaging study, Selvan (2021) suggests that the increase of large model carbon footprint does not translate into proportional accuracy gains. Measuring this impact is a first step for raising awareness and controlling the impact of NLP experiments and operations. Some guidelines were offered in the SustaiNLP workshop[1] to measure impact, and it was suggested that different methods for measuring environmental impact can lead to different conclusions in terms of algorithm efficiency (Cao et al., 2020).

Our goal is to conduct a systematic review of tools available for measuring the impact of NLP tools and to offer a comparative analysis from the perspective of calculated impact measures and usability. We seek to understand the methods implemented by the tools as well the criteria used to assess impact. The contributions of this study are threefold:

- We identify tools available for measuring the environmental impact of NLP experiments
- We characterize impact measurement tools with respect to scope of the impact information provided and usability
- We apply the tools to assess the impact of named entity recognition experiments in order to compare the measurement obtained in two computational set-ups.

2 Environmental impact due to deep learning programs

As for any Information and Communication Technology (ICT) service, a deep learning program impacts several environmental indicators, among which the Global Warming Potential, expressed in terms of GreenHouse Gases (GHGs) emissions, CO_2 equivalent emissions, or carbon footprint. Other indicators include: abiotic resource depletion, blue water shortage, human toxicity... Existing tools for deep learning programs focus on the carbon footprint only.

The environmental impact in terms of carbon footprint needs to account for the entire lifecycle of

[1] https://sites.google.com/view/sustainlp2020/home

ICT equipment from production, through use and finally end of life.

Life Cycle Analysis usually allocates part of the GHG emitted during the production of equipment to the use e.g. when running computer programs, since the equipment was partly produced for the purpose of running programs. For ICT equipment, this phase is difficult to assess because data on GHG emissions during production are not always easily available. In the case of Graphical Processing Unit (GPUs) (or Tensor Processing Unit (TPUs) or equivalents), we could not find any publication or site giving an estimate of the GHG emissions due to the production phase. However, it has to be noted that production can account for a significant part of the total GHG emissions: a French study (Berthoud et al., 2020) on a data center (with Central Processing Unit (CPU) servers only) in Grenoble found that around 40% of the total emissions released during one hour of CPU use were due to the production phase (including emissions due to the equipment alone). Similarly, another recent study (Gupta et al., 2021) reports that the hardware manufacturing and infrastructure accounts for the bulk of the environmental impact of mobile and data center computing equipment, while the impact of operational energy consumption is diminishing.

The end-of-life phase is very difficult to assess for ICT due to lack of data.

To summarize, there are at least four sources of CO_2 equivalent emission sources that should be taken into account to assess the environmental impact of computational experiments: 1/ production of hardware equipment: router, PC, server; 2/ idle use of the hardware; 3/ dynamic use of the hardware; and 4/ end of life of equipment.

3 Tools

Our method for identifying tools and characterizing them relied on a recent study conducting a similar review of annotation tools (Neves and Ševa, 2019). In our study, four authors of this paper contributed to the definition of the criteria and evaluation of the tools. They all have a computer science background with programming experience, and were not involved in the development of any of the tools reviewed or selected.

3.1 Selection of tools

We started from a short list of tools identified by a Working group on the environmental impact of AI in the French group EcoInfo [2] (Experiment Impact tracker, Pyjoules, Carbon tracker).

Then we used snowballing to collect articles citing these tools (according to Google Scholar). For articles published in ArXiv, we also reviewed "related papers" when available. We repeated the process for each newly identified tool we selected.

The goal of this study is to survey and analyze tools that are openly available to the scientific community to evaluate the carbon footprint of natural language processing experiments. Therefore, we selected tools that meet the following criteria:

- freely available
- usable in our programming environment (Mac/linux terminal)
- documented in a scientific publication
- suitable to measure the impact of NLP experiments, such as named entity recognition
- providing a CO_2 equivalent measure for experiments

From our initial shortlist, pyJoules is a python library that monitors the energy consumed by specific device of the host machine such as CPU, RAM, GPU. It is part of the PowerAPI toolkit, which offers solution for measuring energy consumption of software in real time (Bourdon et al., 2013). It was excluded because it does not directly supply a CO_2 equivalent value. Other tools under development were also excluded, when they provided no code or online platform: (Zhang et al., 2020; Shaikh et al., 2021).

Figure 1 presents detailed results of our literature search. A total of 94 publications were obtained from Google Scholar and an additional 20 from ArXiv core related papers. After de-duplication, 85 publications were reviewed. We found that many (N=43) offered opinions or discussion of carbon impact measurement in machine learning, NLP and other fields. Another 27 (represented by the orange flow) described studies that measured the environmental impact of experiments using one of the selected tools. Strubell et al. (2019) presented a study measuring the impact of NLP experiments using methods (nvidia and Intel RAPL system management interface) that are now implemented in some of the selected tools.

[2] https://ecoinfo.cnrs.fr/

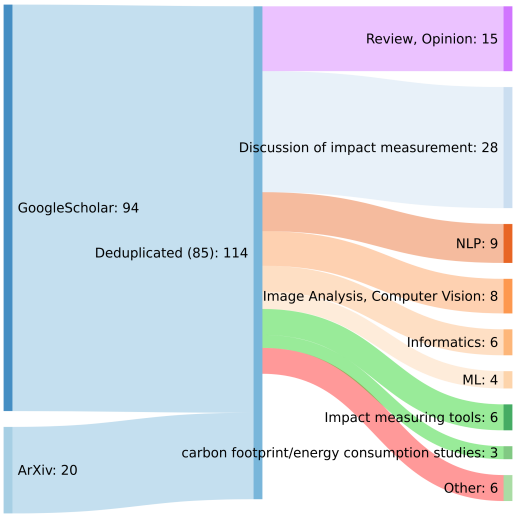

Figure 1: Sankey diagram showing the publications reviewed in our literature search for identifying carbon impact measurement tools.

3.2 List of evaluation criteria

To evaluate the selected impact measurement tools, we defined various criteria to characterize the availability of tools and documentation as well as technical parameters of the tools, including the type of hardware in the scope, the type of measure offered, the detailed information used to assess electricity use by data centers and carbon intensity for electricity production depending on location.

Criteria are split into 4 categories: (a) publication, (b) technical, (c) configuration and (d) functional criteria. Each is presented in detail below.

Publication criteria

P1 - Year of the last publication;

P2 - Citations in Google Scholar (as of 11 May 2021);

P3 - Citations for measuring NLP experiments (as of 11 May 2021).

Technical criteria

T1 - Date of the last version (as of 11 May 2021);

T2 - Availability of the source code;

T3 - Online availability for use;

T4 - Easiness of installation; We evaluated it as "Poor" if we did not manage to install it, "Fair" if we managed to install it but needed system administration access, "Good" if we managed to install it as an ordinary user.

T5 - Quality of the documentation (companion publication or code documentation); We evaluated it as "Poor" if we did not find documentation on the tool, "Fair" if documentation is available, but lacks practical usage details, "Good" if the available documentation addresses usage questions such as parameter settings and country localization.

T6 - Type of license

T7 - Output formats

Configuration criteria

C1 - Local values for carbon intensity: Are local values automatically taken into account or is a global energy mix used?

C2 - Possible (manual) configuration of carbon intensity; Yes if it is possible to configure the carbon intensity without changing the code; No otherwise; We also note whether instructions are provided to the users as to where adequate values can be found.

C3 - Possible (manual) configuration of PUE; Yes if it is possible to configure the PUE without changing the code; No otherwise; We also note whether instructions are provided to the users as to where adequate values can be found.

C4 - Platforms taken into account; which type of equipment is covered by the measurements: PC, server, cloud?

C5 - Other configuration features

Functional criteria

F1 - CO_2 equivalent emission sources taken into account; We consider the following sources, described in section 2: *production*, *idle use*, *dynamic use* and *end of life*.

F2 - Hardware taken into account: does the calculation model account for emissions from data transmission between equipment types as well as from the hardware executing the experiments?

All the tools are supposed to take both CPU and GPU consumption into account, so we did not include this criterion in our analysis.

3.3 List of selected tools

We introduce the selected tools below. Table 1 presents the evaluation of the tools according to the criteria defined in section 3.2.

Green Algorithms (Lannelongue et al., 2021) is an online tool developed by researchers in the UK that calculates the energy consumption and carbon footprint of computer use based on information supplied by the user in a web interface: runtime, number of cores, memory requested, type of platform used (PC, local server, cloud computing), type of cores, location.

ML CO2 Impact (Lacoste et al., 2019) is an online tool developed by researchers in Canada that calculates the energy consumption and carbon footprint of computer use based on user supplied information including hardware, runtime, cloud provider and location of the computing facilities operated. We are aware that a new version of the tool is being developed under the umbrella of the *Code Carbon* [3] initiative. However, it is not yet described in a scientific publication so we have decided to evaluate ML CO2 which has been used by the NLP research community.

Energy Usage (Lottick et al., 2019) was developed by researchers in the United States with the goal of improving accountability in machine learning research. Experiment impact tracker (Henderson et al., 2020) was developed by researchers in north America to assist researchers in measuring and reporting the impact of their machine learning experiments. Carbon tracker (Anthony et al., 2020) was developed by researchers from Denmark for tracking and predicting the energy consumption and carbon footprint of training deep learning models. These three tools are python packages that get the energy consumption of a machine learning program via GPU, CPU and DRAM information.

Cumulator (Tristan Trebaol and Ghadikolaei, 2020) is also a python package developed in Switzerland. Its estimates the energy consumption of computation based on runtime, GPU load and carbon intensity, with a fixed value for consumption of a typical GPU. It also estimates the energy consumption of communication based on the file sizes and the 1 byte model from The Shift Project (The Shift Project, 2018).

[3] https://codecarbon.io/

4 Use case: measuring the impact of named entity recognition

In this section, we present experiments for a typical NLP task, named entity recognition (NER). Experiments are conducted on two computational set-ups: (1) the use of a server within the laboratory (equipped with 2 GPUs Nvidia GeForce GTX 1080 Ti) and (2) the use of an external shared computer facility (equipped with 43 GPUs including GPU Nvidia Tesla V100). We hypothesize that this type of set-up can be available to NLP researchers, and that it is relevant to document the implications of choosing one or the other for a set of experiments.

We measure the impact of the experiments using the selected tools and compare the carbon footprint across tools and computational set-ups.

4.1 Named Entity Recognition on QUAERO benchmark corpora

NER methods. Many Named Entity Recognition (NER) models focus on identifying flat entities (Collobert et al., 2011; Lample et al., 2016; Ma and Hovy, 2016; Peters et al., 2018; Luoma and Pyysalo, 2020) based on a sequence labeling approach. However, to address the need for the extraction of nested entities, an increasing number of models do take nested entities into account as well (Alex et al., 2007; Lu and Roth, 2015; Ju et al., 2018; Straková et al., 2019; Yu et al., 2020; Li, 2021). Nested entities are embedded named entities included in other entities. To reflect current needs of entity extraction, we choose to evaluate the energy consumption of two deep learning neuronal models, one that addresses flat entity recognition (Ma and Hovy, 2016) and one that addresses both flat and nested entity recognition, introduced by (Yu et al., 2020). (Yu et al., 2020) adapt the biaffine dependency parsing model of (Dozat and Manning, 2017) to Named Entity Recognition by reformulating this task as the task of identifying start and end indices and associating a category to the span defined by these pairs. The biaffine model is used on top of a multi-layer BiLSTM to assign score to all possibles spans. To reproduce this model, we encode words using the pre-trained Language model CamemBERT$_{\text{BASE}}$ (Martin et al., 2019) provided by the Transformers library[4] from Hugging Face Inc. (Wolf et al., 2020) and we follow the same strategy by using CNN to encode the character-based word embeddings.

[4] https://huggingface.co/transformers/

	Carbon Tracker (Anthony et al., 2020)	**Green Algorithms** (Lannelongue et al., 2021)	**Experiment Impact Tracker** (Henderson et al., 2020)	**ML CO2 Impact** (Lacoste et al., 2019)	**energy usage** (Lottick et al., 2019)	**Cumulator** (Tristan Trebaol and Ghadikolaei, 2020)
P1	2020	2021	2020	2019	2019	2020
P2	18	4	33	35	4	0
P3	1 (Parcollet and Ravanelli, 2021)	1 (Liu et al., 2021)	3 (Cao et al., 2020; Prasanna et al., 2020; Peng et al., 2021)	4 (Sarti, 2020; Selby et al., 2021; Chaudhary et al., 2020; Gencoglu, 2020)	0	0
T1	Dec 8, 2020	Dec 17, 2020	April 29, 2021	May 4, 2021	July 10, 2020	April 29, 2021
T2	Yes	Yes	Yes	Yes	Yes	Yes
T3	No	Yes (online)	No	Yes (online)	No	No
T4	Good	No install needed	Fair	No install needed	Poor	Good
T5	Good	Fair	Fair	Fair	Fair	Good
T6	MIT	CC-BY-4.0	MIT	MIT	Apache	MIT
T7	Generates a statement to report results	Generates a statement to report the results	Generates a statement and graphs to report results	Generates text and LaTeXcode to report the results	Generates text and pdf reports	Generates a text report
C1	Yes: carbon intensity from Energi data service for Denmark, Carbon intensity API for the UK, CO2 signal API otherwise, and European Environment Agency for the default value	Yes, carbon intensity from carbonfootprint	Yes, carbon intensity from electricitymap	Yes, pointers supplied to user, including electricitymap	Yes, carbon intensity from U.S. Energy Information Administration data and U.S. Environmental Protection Agency eGRID data	No
C2	No	No	No	Yes	No	No, but indications are given in the documentation about how to change the default value
C3	No, default value of PUE = 1.67 (2019)	No, default value of PUE = 1.67 (2019)	Yes. Default PUE (1.58) can be adjusted	Partly, for Google, Amazon, Azure cloud providers.	No PUE used but PSU loss can be set	No PUE used
C4	Install dependent	PC, local server, cloud	Install dependent	3 specific providers, private infrastructure	Install dependent	Install dependent
C5	Consumption prediction based on a number of epochs, monitoring of chosen components, conversion to interpretable numbers...	Pragmatic Scaling Factor to take into account the number of experiments, conversion to interpretable numbers, comparison with other locations	Asserting certain hardware	No	Year for the data, comparison with other locations	No
F1	Dynamic use	Dynamic use	Dynamic use	Dynamic use	Dynamic use	Dynamic use
F2	Hardware only	Hardware only	Hardware only	Hardware only	Hardware only	Hardware and communication

Table 1: Evaluation of the tools according to the publication (P), technical (T), configuration (C) and functional (F) criteria.

Experimental set-up. The following configuration is used: 1 core is used; GTX 1080 Ti GPUs are used on the lab server while Tesla V100 GPUs are used on the computing facility; the memory used is 11 GB on the server, and 32GB on the facility; 20 CPUs are used on the facility; the experiments were conducted in France. We documented France as the location for experiments in the impact measurement tools as directed by the tools' documentation.

Datasets. We used two benchmark datasets for Named Entity Recognition in French: the QUAERO Broadcast News Extended Named Entity dataset (Galibert et al., 2010) and the QUAERO French Med dataset (Névéol et al., 2014), which contains the EMEA and MEDLINE subsets. We selected these datasets because they are suitable for the task at hand, illustrate different scales of trainting data (the Press corpus is much larger than the medical corpus) while remaining of modest size, and are freely available for non-commercial use.

Table 2 presents descriptive statistics including details about nested entities for the datasets.

	QUAERO French News	QUAERO French Med EMEA	QUAERO French Med MEDLINE
Documents	167	38	2,498
Entity Types	5	10	10
Tokens	1,347,368	40,257	31,926
Entities	79,632	7,159	9,074
Unique entities	19,876	1,880	5,895
Nested entities	0	1,009	2,280
% Nested	0 %	14,27 %	25,31 %
Max Depth	1	4	4

Table 2: Corpus Statistics.

4.2 Results

Table 4 presents the results of the experiments, with the CO_2 equivalent measures for training each model. To further understand potential differences in CO_2 measures we also report the corresponding energy consumption in Table 3.

The CO_2 equivalent emissions measured for a same experiment greatly depending on the measuring tool used. We analyzed the computational set-up and tool configurations in an attempt to better understand the observed variations.

Carbon intensity We noticed that carbon tracker used the average carbon intensity for EU-28 in 2017 (294.21 gCO_2eq/kWh) instead of the French value (around 30 to 40 gCO_2eq/kWh according to electricityMap), which overestimates the CO_2 equivalent cost. Green Algorithms, which uses the 2020 values from electricityMap, gives 39 gCO_2eq/kWh. Experiment impact tracker uses the 2018 electricityMap value, which gives a 47.60 gCO_2eq/kWh for France. Energy usage relies on international energy mix data from the U.S. Energy Information Administration data for the year 2016, and assumed carbon equivalencies by energy type. The value for France thus seems to be 424 gCO_2eq/kWh. ML CO2 Impact was used with the default value, 432 gCO_2/kWh. We investigated the data sources provided by the tools to search for a more precise value for France, but the values for Carbon intensity varied a lot: 53 gCO_2/kWh on Carbon footprint when the given link was followed, leading to the 2018 emissions; with the most recent data available, from 2020, the carbon intensity for France is 38,95. On electricityMap, the value at the time of the experiments was 31 gCO_2/kWh. The European Commission link again gives different values, according to the kind of electricity considered, and based on 2013 values.

The carbon intensity values is thus very different even when considering the same country.

Hardware On the computing facility, we used Tesla V100-PCIE-32GB GPUs. However, the hardware options offered by the online algorithms did not exactly correspond to the equipment we used. For example ML CO2 Impact offers only V100-PCIE-16GB or V100-SXM2-32GB, so we reported the results for V100-SXM2-32GB. In Green algorithms, the only GPU option available is Tesla V100. This may lead to a lack of precision on the results.

Green algorithms currently does not enable to input both CPU and GPU usage, although a forthcoming version should include this possibility.

Precision of results The NER experiments conducted were designed to test the impact measurement tools without causing excessive environmental impact. We used modest sized datasets requiring short training times, which posed problems for some tools. For example ML CO2 Impact displayed 0 values for several experiments, so we used the formula provided in the publication to obtain a more precise measurement.

5 Discussion

Measuring carbon footprint is a novel undertaking Table 1 shows that the availability of tools

		CO_2 equivalent (g.)						Runtime (mins.)	NER metrics		
		CT	EIT	EU	Cu	MLCI	GA		P	R	F
NER (Yu et al., 2020)	**French Press**										
	Server	237.96	78	0.496	302	290	350.15	163:39	87.49	74.85	80.68
	Facility	161.16	48	0.979	222	250	260.26	118:04	88.05	74.71	80.83
	EMEA										
	Server	9.70	30	0,00131	19	20	16.67	9:31	73.78	59.74	66.02
	Facility	8.07	1	0,002	13.7	10	14.31	6:51	77.58	58.71	66.84
	MEDLINE										
	Server	13.44	30	0,00128	26.1	20	20.68	11:55	66.62	62.11	64.28
	Facility	10.50	1	0,00259	19.4	20	20.03	9:11	79.73	78.35	78.98
NER (Ma and Hovy, 2016)	**French Press**										
	Server	87.62	12	5.1	100.04	125	104.40	58:30	78.49	69.77	73.87
	Facility	46.43	6	2.87	79.05	99	102.08	46:44	80.75	70.67	75.38
	EMEA										
	Server	2.23	0.004	0.117	4.31	0	3.83	02:14	61.77	50.27	55.43
	Facility	2.28	0	0.151	3.23	0	4.99	02:27	57.46	51.98	54.58
	MEDLINE										
	Server	2.99	0	0.137	5.20	0	5.57	03:11	43.97	41.08	42.47
	Facility	2.74	0	0	0.176	0	5.67	02:58	52.39	36.68	43.15

Table 3: Results of NER experiments. The upper part of the table presents the results obtained with an implementation of the method by (Yu et al., 2020) while the bottom part presents the results obtained with an implementation of the method by (Ma and Hovy, 2016). The CO_2 equivalent measures are reported according to the six selected tools in this study, Carbon Tracker (CT), Green Algorithms (GA), Experiment Impact Tracker (EIT), ML CO2 Impact (MLCI), Energy Use (EU) and Cumulator (Cu).

		Energy consumption (kWh)					
		CT	EIT	EU	Cu	MLCI	GA
NER (Yu et al., 2020)	**French Press**						
	Server	0.809	1.399	0,00117	n/a	0.68	1.38
	Facility	0.548	0.865	0,00231	n/a	0.59	1.03
	EMEA						
	Server	0.033	0.053	0,0000034	n/a	0.04	0.07
	Facility	0.027	0.017	0,0000047	n/a	0.03	0.06
	MEDLINE						
	Server	0.046	0.045	0,0000030	n/a	0.05	0.08
	Facility	0.036	0.021	0,0000061	n/a	0.05	0.08
NER (Ma and Hovy, 2016)	**French Press**						
	Server	0.298	0.209	0.012	n/a	0.29	0.41
	Facility	0.158	0.102	0.0068	n/a	0.23	0.40
	EMEA						
	Server	0.0072	0.007	0.00028	n/a	0.01	0.02
	Facility	0.0078	0.004	0.00036	n/a	0.01	0.02
	MEDLINE						
	Server	0.010	0.007	0.00032	n/a	0.015	0.02
	Facility	0.0094	0.005	0.0004	n/a	0.015	0.02

Table 4: Energy consumption in kWh for each method and experimental condition. The upper part of the table presents the results obtained with an implementation of the method by (Yu et al., 2020) while the bottom part presents the results obtained with an implementation of the method by (Ma and Hovy, 2016). The measures are reported according to the six selected tools in this study, Carbon Tracker (CT), Green Algorithms (GA), Experiment Impact Tracker (EIT), ML CO2 Impact (MLCI), Energy Use (EU) and Cumulator (Cu).

for measuring carbon impact of experiments is quite recent. Furthermore, these tools have been moderately used in the field of NLP (we identified 9 studies in total). However, we can note that the interest in producing impact measures is widespread as the six tools reviewed were developed by re-

searchers in different countries across Europe and North America.

Differences in carbon footprint measurements Miozzo et al. (2021) found the emissions computed by Green algorithms to be higher than those computed by ML CO2 Impact. This is explained by the details of the *functional criteria* reported in table 1, which show that key elements such as carbon intensity values or PUE differ across the tools. The results from table 4 also illustrate the differences.

The CO_2 equivalent emissions obtained from ML CO2 Impact and Green Algorithms are higher than those returned by other tools. This is consistent with our expectation because GA and MLCI do not perform direct measurements of the energy consumption but estimate it based on user supplied information. Carbon Tracker and Experiment Impact Tracker, which are based on the same calculations, produce similar impact measures. We observe that impact measures, in terms of energy consumption as well as CO_2 emissions, are generally lower for the computing facility compared to the local server. This can probably be explained by the difference in equipment (e.g. type of GPUs).

The measures obtained from Energy Usage are much lower than those obtained from the other tools. After investigation, it seems that these results do not take into account the GPU consumption, due to a possible bug in the tool, which was reported to the code authors.

Carbon footprint is underestimated All the tools reviewed in this study only evaluate the carbon footprint of NLP experiments based on energy consumption during the dynamic use phase of equipment. Emissions resulting from the production and end-of-life phases are unaccounted or partially accounted. In addition, we can note that setting up experiments also requires some upstream testing of configurations and model parameters that are often not accounted in the measures. Green Algorithm is the only tool in our selection that brings the users attention to this source of emission with the "pragmatic scaling factor". For these reasons, the measures of CO_2 equivalent obtained with these tools underestimate the actual carbon footprint.

NER performance The performance of the systems in terms of precision, recall and F-measure is above the median and average of participations in the 2016 CLEF eHealth task where the QUAERO French Med dataset was used as a benchmark (Névéol et al., 2016)[5]. As expected, the performance obtained is higher with the more recent nested entity extraction tool (Yu et al., 2020) vs. flat entity extraction tool (Ma and Hovy, 2016). This comes at the cost of higher environmental impact as measured by all tools. We can also note that the state of the art on this dataset remains the low-carbon cost dictionary method submitted to the CLEF eHealth shared task by the Erasmus team (Van Mulligen et al., 2016).

Which tool should be used for measuring carbon footprint of NLP experiments? The online tools (Green Algorithms and ML CO2 impact) are very convenient to use as no installation is necessary. Since they are used separately from running the experiments, an estimate of experiment impact can be obtained after running the experiment. However, some of the information to be supplied is not easy to figure out, such as "memory requirement" (GA) or "carbon intensity" (MLCI). In our experience, the use of the python packages tracking real-time energy usage (Carbon Tracker, Experiment Impact Tracker and Energy Usage) required special permission to read RAPL results, even with through powercap access, so admin assistance was needed to use the tools.

Also, the short training times in the NER experiments yielded impact measures of 0 from some of the tools. This suggests that these tools are not sensitive enough to measure small impacts and may be intended for use on higher impact experiments.

6 Conclusions

In this paper, we have conducted a survey of the literature to identify tools for measuring the environmental impact of NLP experiments. We characterized six tools and evaluated them on a sample named entity recognition task. The measures obtained vary significantly and only account for one in four sources of carbon emissions. More work is needed to better understand the differences in measurement between the tools and to account for sources of carbon emissions other than the dynamic use of hardware equipment: production, idle use, end of life.

[5]We note that while the results of the nested entity extraction tool (Yu et al., 2020) are directly comparable to the shared task results, those of the flat entity extraction tool (Ma and Hovy, 2016) are not because 14-25% of nested entities are not taken into account.

Acknowledgments

The sankey diagram presented in Figure 1 was prepared using http://www.sankeymatic.com/build/.

Author contributions

Anne-Laure Ligozat designed the study, developed the list of criteria and wrote a first draft of the manuscript. Aurélie Névéol conducted the literature search and contributed to manuscript drafting. Nesrine Bannour and Sahar Ghannay conducted the NER experiments and impact measurements. All authors contributed to the evaluation of tools.

References

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Biological, translational, and clinical language processing*, pages 65–72.

Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. 2020. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. In *ICML Workshop on "Challenges in Deploying and monitoring Machine Learning Systems"*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? 🦜. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Francoise Berthoud, Bruno Bzeznik, Nicolas Gibelin, Myriam Laurens, Cyrille Bonamy, Maxence Morel, and Xavier Schwindenhammer. 2020. Estimation de l'empreinte carbone d'une heure.coeur de calcul. Research report, UGA - Université Grenoble Alpes ; CNRS ; INP Grenoble ; INRIA.

Aurélien Bourdon, Adel Noureddine, Romain Rouvoy, and Lionel Seinturier. 2013. PowerAPI: A software library to monitor the energy consumed at the process-level. *ERCIM News*, 2013(92).

Qingqing Cao, Aruna Balasubramanian, and Niranjan Balasubramanian. 2020. Towards accurate and reliable energy measurement of NLP models. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 141–148, Online. Association for Computational Linguistics.

Yatin Chaudhary, Pankaj Gupta, Khushbu Saxena, Vivek Kulkarni, Thomas Runkler, and Hinrich Schütze. 2020. TopicBERT for energy efficient document classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1682–1690, Online. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12:2493–2537.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR*.

Olivier Galibert, Ludovic Quintard, Sophie Rosset, Pierre Zweigenbaum, Claire Nédellec, Sophie Aubin, Laurent Gillard, Jean-Pierre Raysz, Delphine Pois, Xavier Tannier, Louise Deléger, and Dominique Laurent. 2010. Named and specific entity detection in varied data: The quæro named entity baseline evaluation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Oguzhan Gencoglu. 2020. Large-scale, language-agnostic discourse classification of tweets during covid-19. *Machine Learning and Knowledge Extraction*, 2(4):603–616.

Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S. Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. 2021. Chasing carbon: The elusive environmental footprint of computing. In *IEEE International Symposium on High-Performance Computer Architecture (HPCA 2021)*. IEEE.

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):1–43.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. In *Climate Change workshop, NeurIPS 2019*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL*.

Loïc Lannelongue, Jason Grealey, and Michael Inouye. 2021. Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, page 2100707.

B. Li. 2021. Named entity recognition in the style of object detection. *ArXiv*, abs/2101.11122.

Leo Z. Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021. Probing across time: What does roberta know and when? *CoRR*, abs/2104.07885.

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 NeurIPS 2019*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867.

Jouni Luoma and Sampo Pyysalo. 2020. Exploring Cross-sentence Contexts for Named Entity Recognition with BERT. In *COLING*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219. Association for Computational Linguistics.

Marco Miozzo, Zoraze Ali, Lorenza Giupponi, and Paolo Dini. 2021. Distributed and multi-task learning at the edge for energy efficient radio access networks. *IEEE Access*, 9:12491–12505.

Aurélie Névéol, K Bretonnel Cohen, Cyril Grouin, Thierry Hamon, Thomas Lavergne, Liadh Kelly, Lorraine Goeuriot, Grégoire Rey, Aude Robert, Xavier Tannier, and Pierre Zweigenbaum. 2016. Clinical information extraction at the clef ehealth evaluation lab 2016. In *CEUR workshop proceedings*, volume 1609, pages 28–42.

Aurélie Névéol, Cyril Grouin, Jeremy Leixa, Sophie Rosset, and Pierre Zweigenbaum. 2014. The QUAERO French medical corpus: A resource for medical entity recognition and normalization. In *Proc. BioTextM*.

Mariana Neves and Jurica Ševa. 2019. An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*, 22(1):146–163.

Titouan Parcollet and Mirco Ravanelli. 2021. The Energy and Carbon Footprint of Training End-to-End Speech Recognizers. Working paper or preprint.

Xutan Peng, Guanyi Chen, Chenghua Lin, and Mark Stevenson. 2021. Highly efficient knowledge graph embedding learning with orthogonal procrustes analysis. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3208–3229, Online. Association for Computational Linguistics.

Gabriele Sarti. 2020. Interpreting neural language models for linguistic complexity assessment. Master's thesis, University of Trieste.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2019. Green AI. *Communications of the ACM*. Published in 2020.

Kira A. Selby, Yinong Wang, Ruizhe Wang, Peyman Passban, Ahmad Rashid, Mehdi Rezagholizadeh, and Pascal Poupart. 2021. Robust embeddings via distributions. *CoRR*, abs/2104.08420.

Raghavendra Selvan. 2021. Carbon footprint driven deep learning model selection for medical imaging. Https://openreview.net/forum?id=1TPRpNyyj2L.

Omar Shaikh, Jon Saad-Falcon, Austin P Wright, Nilaksh Das, Scott Freitas, Omar Isaac Asensio, and Duen Horng Chau. 2021. Energyvis: Interactively tracking and exploring energy consumption for ml models. *arXiv preprint arXiv:2103.16435*.

Jana Straková, Milan Straka, and Jan Hajič. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331. Association for Computational Linguistics.

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*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

The Shift Project. 2018. Lean ICT: Towards Digital Sobriety. Technical report, The Shift Project. Directed by Hugues Ferreboeuf.

Julien Tourille, Matthieu Doutreligne, Olivier Ferret, Nicolas Paris, Aurélie Névéol, and Xavier Tannier. 2018. Evaluation of a sequence tagging tool for biomedical texts. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 193–203.

Martin Jaggi Tristan Trebaol, Mary-Anne Hartley and Hossein Shokri Ghadikolaei. 2020. A tool to quantify and report the carbon footprint of machine learning computations and communication in academia and healthcare. *Infoscience EPFL: record 278189*.

Erik M Van Mulligen, Zubair Afzal, Saber Akhondi, Dang Vo, and Jan Kors. 2016. Erasmus mc at clef ehealth 2016: Concept recognition and coding in french texts. In *CEUR workshop proceedings*, volume 1609, pages 171–178.

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*, pages 38–45, Online. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476. Association for Computational Linguistics.

Huaizheng Zhang, Yizheng Huang, Yonggang Wen, Jianxiong Yin, and Kyle Guan. 2020. No more 996: Understanding deep learning inference serving with an automatic benchmarking system. *arXiv preprint arXiv:2011.02327*.