AI and FOSS: looking for founding principles

Roberto Di Cosmo
Computer Science Professor
Director, Software Heritage
https://dicosmo.org
@rdicosmo
Perspectives on AI and code

1. The user
2. The software commons
3. The developers (next time :-))
Preserving user freedom in an AI world

From the April 2018 LLW workshop in Barcelona
(thanks to Andy Wilson)

How can we preserve the “user freedom”, in the spirit of FOSS, in an “automated decision making world”?

- Well into the debate about “ethical principles” for AI
- Way before the Generative AI explosion of interest
Path 1: Explainable AI

**Principle:** “Any end user that is the object of an automated decision should be given a human understandable explanation of why the decision was made.”

*It’s the same that we expect from a judgement: not the sequence of neuron firings in the head of a judge, but the “rationale” of the decision.*

This seems to be the golden standard *from a user point of view.*

*UPDATE: needs to be refined* with generative AI (what is an explanation here?)
Path 2: Accountability and Transparency

**Principle:** “When a human understandable explanation cannot be obtained, it is in general not possible to assess the outcome of an automated decision just by looking at it. Hence we believe that the whole decision making process should be transparent, and accountable.”

In the case of machine learning, this includes, in particular,

- the (source code of the) software used for the processing
- the trained machine model(s)
- the data used in the training
- and all information necessary to perform independent experiments using all the above

This is a plan B for a user, but seems to be the gold standard for research in ML.

UPDATE: very hot topic with LLMs, see later (also AI act)
Path 3: Ethics

**Principle:** “When we cannot control the outcome of the use of a technology, or we fear we do not sufficiently understand the consequences, we should just refrain from using it.”

This is a *last resort for a user*, but has been the gold standard *for research in biotechnology, for decades*.

**Big question:** who makes the decisions?

See e.g. [the UNESCO recommendation on ethics for AI](https://www.unesco.org/new/en/communication-and-information/communication/our-work/strategy/ai/s/ai-guidelines/)


Preserving software commons in a (Gen)AI world

Software commons are massively used for building Large Language Models.

Independently of what we do, there is no turning back.

The real question is how they should be built and whom they should benefit.

Let’s have a candid look around us.
Closed model APIs

OpenAI

Gemini

ANTHROPIC

Open model weights

Meta

Mistral.AI

deepseek

This slide is courtesy of Leandro Von Werra and Harm de Vries
Closed model APIs

- Model weights not available
  - Can’t run the model locally
  - Can’t inspect the model’s representations
  - Limits fine-tuning abilities

And more:
  - limits user freedom (personal data leakage)

Open model weights

This slide is courtesy of Leandro Von Werra and Harm de Vries
Closed model APIs

- Model weights not available
  - Can't run the model locally
  - Can't inspect the model's representations
  - Limits fine-tuning abilities

And more:
- limits user freedom (personal data leakage)

Open model weights

- Training data is not disclosed
  - Content creators don't know if their data is used
  - There’s no way to remove it
  - Can't inspect data for biases
  - Potential benchmark contamination

This is not what “open” should mean.

Can we change all this? How?

This slide is courtesy of Leandro Von Werra and Harm de Vries
A window of opportunity

https://a16z.com/generative-ai-enterprise-2024/

Companies want “open source” models (sorry Stefano, not my words!), but...

LLMs follow a winners take all dynamics!
The source code opportunity

Largest archive of source code
digital commons built since 2015

Cultural Heritage  Industry  Research  Public Administration

Source files 17,567,724,625
Commits 3,730,352,827
Projects 274,163,348

Directories 14,101,884,066
Authors 68,895,808
Releases 81,171,066

500+ code hosting platforms
All versions, full development history
In a single giant Merkle Graph

Ensures availability
 Guarantees integrity
 Enables traceability

Unique dataset for machine learning,
an infrastructure for transparency and accountability

- 35 × 10⁹ nodes
- 500 × 10⁹ edges
~ 2 PB storage
Looking for founding principles at Software Heritage

Principles

1. Knowledge derived from the Software Heritage archive must be given back to humanity, rather than monopolized for private gain. The resulting machine learning models must be made available under a suitable open license, together with the documentation and toolings needed to use them.

2. The initial training data extracted from the Software Heritage archive must be fully and precisely identified by, for example, publishing the corresponding SWHID identifiers (note that, in the context of Software Heritage, public availability of the initial training data is a given: anyone can obtain it from the archive). This will enable use cases such as: studying biases (fairness), verifying if a code of interest was present in the training data (transparency), and providing appropriate attribution when generated code bears resemblance to training data (credit), among others.

3. Mechanisms should be established, where possible, for authors to exclude their archived code from the training inputs before model training begins.

Question: are we asking too much?
Findings from **BigCode: The Stack v2** and **StarCoder2**

**Released February 28th 2024**

Yes one can build **the best open LLM for code available** while fully adhering to the Software Heritage principles for responsible LLMs, ...

and even more: **the full training pipeline is made public too!**
The Stack v2

Data collection pipeline fully open and transparent built by BigCode

Software Heritage

select only GitHub repos
license filtering

Raw Dataset
67.5 TB of data
near-deduplication
32.1 TB of data
Language selection, filtering, preprocessing
2.4 TB
775B tokens

GH Issues, PRs, and other high-quality data sources
140B tokens

140B tokens
915B tokens

This slide is courtesy of Leandro Von Werra and Harm de Vries
I found my (L)GPL code in your dataset!

@github copilot, with "public code" blocked, emits large chunks of my copyrighted code, with no attribution, no LGPL license. For example, the simple prompt "sparse matrix transpose, cs" produces my cs_transpose in CSparse. My code on left, github on right. Not OK.


Ethical Concerns of Code Generation Through Artificial Intelligence

By Tim Davis and Siva Rajamanickam

Machine learning models that are trained on large corpuses of text, images, and source code are becoming increasingly common. Such models—which are either freely available or accessible for a fee—can then generate their own text, images, and source code. The unprecedented pace of development and adoption of these tools is quite different from the traditional mathematical software development life cycle. In addition, developers are creating large language models (LLMs) for text summarization as well as caption and prompt generation. LLMs are fine-tuned on source code, such as in OpenAI Codex, which yields models that can interactively generate code with minimal prompting. For example, a prompt like "sort an array" produces code one line at a time that a programmer can then either choose to accept or use to generate a match for an entire sort routine.
Beware of false positives: not everything is copyrightable, e.g. boilerplate, or purely functional code like this one!

Lessons learned

**Principles**

1. Knowledge derived from the Software Heritage archive must be given back to humanity, rather than monopolized for private gain. The resulting machine learning models must be made available under a suitable open license, together with the documentation and toolings needed to use them.
2. The initial training data extracted from the Software Heritage archive must be fully and precisely identified by, for example, publishing the corresponding SWHID identifiers (note that, in the context of Software Heritage, public availability of the initial training data is a given: anyone can obtain it from the archive). This will enable use cases such as: studying biases (fairness), verifying if a code of interest was present in the training data (transparency), and providing appropriate attribution when generated code bears resemblance to training data (credit), among others.
3. Mechanisms should be established, where possible, for authors to exclude their archived code from the training inputs before model training begins.

**Transparency is easy**: SWHID (undergoing ISO standardisation) and Software Heritage N.B. : may be mandated by regulations!

**Opt out is complex**: who is the real right owner? (similar issues to license compliance)

- Building the training set is complex: e.g. includes license compliance alike work at massive scale
- Generating attribution information on model output is more complex than license compliance

We need a **global mutualised effort** to ensure fully open models will succeed!

**Spoiler alert: Software Heritage is engaging in this effort.**
Could we ask for more?

Principles

1. Knowledge derived from the Software Heritage archive must be given back to humanity, rather than monopolized for private gain. The resulting machine learning models must be made available under a suitable open license, together with the documentation and toolings needed to use them.

2. The initial training data extracted from the Software Heritage archive must be fully and precisely identified by, for example, publishing the corresponding SWHID identifiers (note that, in the context of Software Heritage, public availability of the initial training data is a given: anyone can obtain it from the archive). This will enable use cases such as: studying biases (fairness), verifying if a code of interest was present in the training data (transparency), and providing appropriate attribution when generated code bears resemblance to training data (credit), among others.

3. Mechanisms should be established, where possible, for authors to exclude their archived code from the training inputs before model training begins.

Pre GenAI principles

- the (source code of the) software used for the processing
- the trained machine model(s)
- the data used in the training
- and all information necessary to perform independent experiments using all the above

Other ideas...

- Limit use to specific licences?
- Tagging / attribution on gen code?

Ok for StarCoder2 and OlmO, ... what about all the others?

Is this in the interest of FLOSS?
Lessons learned, cont’d

Fully open models…

Pros Full transparency on all stages of model development
- External inputs to the project
- Scientific reproducibility

Cons Resource overhead
- Legal risks of data transparency
- Giving away competitive edge
- Code and data maintenance

The playing field today is tilted in favour of closed/open weights models…

We need a global effort to make open models succeed!
A plurality of actors we can engage with

European AI Office

GPAI / THE GLOBAL PARTNERSHIP ON ARTIFICIAL INTELLIGENCE

The AI Alliance

United Nations | AI Advisory Body

Global AI Ethics and Governance Observatory