ZOOOM Policy Briefs

#1

ZOOOM

Open Source AI:

Building Blocks for a Definition

Authors: Ivo Emanuilov, Jutta Suksi

Published: November 2023, Brussels

Funded by the European Union

Grant agreement No. 101070077
Duration: 10/2022 – 09/2024
Funded by the European Union. Views and opinions expressed are however those of the ZOOOM Consortium and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them.

ZOOOM

Contents

Executive Summary

This policy brief defines Open Source AI from the viewpoint of intellectual property rights. For that purpose, this brief presents a walkthrough of the different phases of building AI, specifically machine learning, the essential components in the process, and the legal and organisational implications thereof.

As AI is a diverse collection of components, it raises the question whether and how copyright subsists in one or more of these different objects in the course of different phases of the pipeline. We address this issue from the following points of view:

- What type of activities does the EU text and data mining exception cover?
- What is the temporal limitation of the EU text and data mining exception and does it cover memorised data?
- Is numerical data copyrightable?
- Do I need to comply with the licence conditions when my model is used?
- Do I need to comply with the licence conditions when using open source code as training data?

In addition, we assess, how intellectual property law treats hybrid intellectual property.

We offer three building blocks for a future definition of Open Source AI, namely transparency, enablement and reproducibility:

1. Transparency: disclosure of details about the composition of training data sets, details about the data structures, architecture and algorithms, access to neural network weights etc.

2. Enablement: disclosure of sufficient details about the building of a model to enable anyone to rebuild the model, provided they have access to the required computational resources, as identified by the community developing the AI.

3. Reproducibility: development practices that create an independently-verifiable path from the training data to model inference.

These three building blocks should unlock the opportunities of open source in the domain of AI and we expect that they would also facilitate comprehension of AI as protected and licensable subject matter.

Introduction

Our aim is to define Open Source AI from the viewpoint of intellectual property rights. For that purpose, this brief presents a walkthrough of the different phases of building AI, specifically machine learning, the essential components in the process, and the legal and organisational implications thereof. We offer three building blocks for a future definition of Open Source AI, namely transparency, enablement and reproducibility. These three building blocks should unlock the opportunities of open source in the domain of AI and we expect that they would also facilitate comprehension of AI as protected and licensable subject matter.

The Lifecycle of AI

If we look at the pipeline of building AI, there is the training data and an untrained model which needs to be trained and then applied to new, unseen data to get a useful output. On this very general level the process could be visualized as follows:

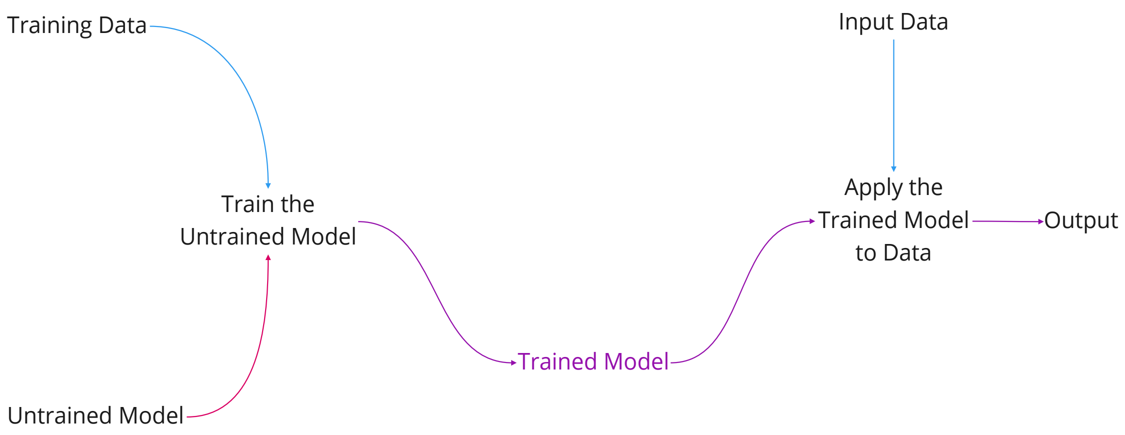


Figure 1. AI lifecycle (general overview)

When we look at the training data in more detail, we realise there are also preliminary phases involved. Before we get to the training data, we need to access, collect and prepare the data. Many of the large language models today rely on unsupervised learning on massive datasets, but in many cases, we would have to also label some of the data.

We then need to build the untrained model which involves choosing a model architecture and an optimisation strategy. Furthermore, there are other elements that belong neither to the side of data nor to the side of the model. For instance, we would often have pretrained models and different forms of knowledge representations.

Additionally, the trained model in itself is not a simple object. There are several components. For example, weights are randomly initialised at first, followed by a process of 'improving' them as determined by the neural network's loss function. In addition to that, we have biases, network topology, activation functions, and representations. The weights and biases are the learnable parameters of neural networks.

The final step in this pipeline is the generation of output. This can be an answer to a question or the generation of a new object, such as image, code or video. The answer itself can also be data.

When collected into one visualisation, these components and phases would look like this:

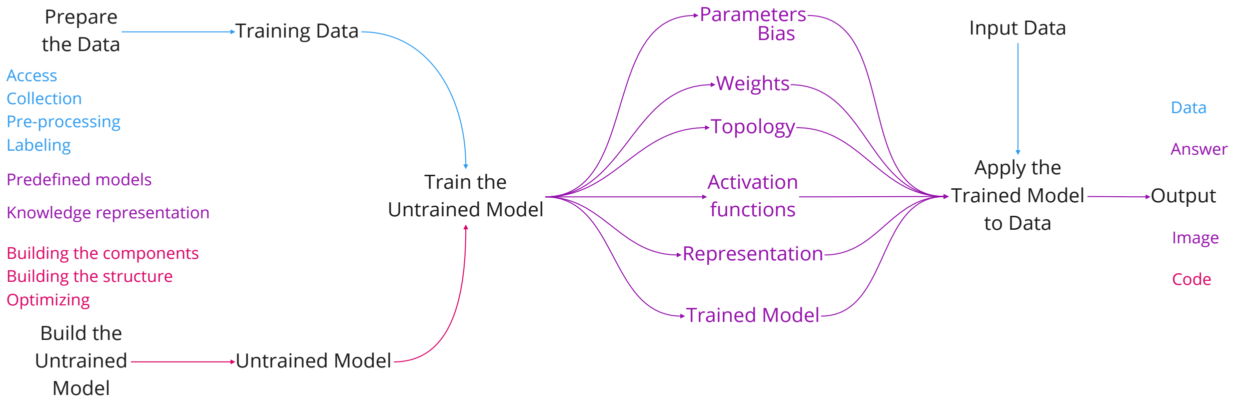


Figure 2. AI lifecycle (detailed overview)

As a diverse collection of components, this raises the question whether and how copyright subsists in one or more of these different objects in the course of different phases of the pipeline.

From Code to Data… to Code: Is There Copyright in Data?

How concerned should we be about copyright that may subsist in the datasets? The answer is critical because if we find that copyright subsists in the dataset, then this means that whatever happens to it in the learning process could result, for example, in a derivative work.

In this section, we discuss the subsistence of copyright in the trait between the training data and output. In the lifecycle of AI, this can be traced to the following path:

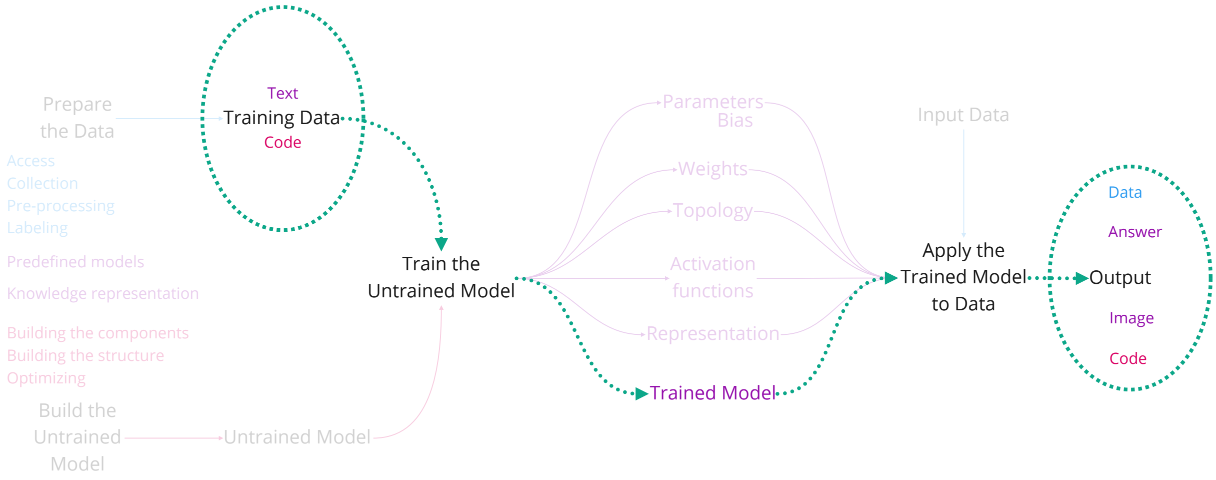
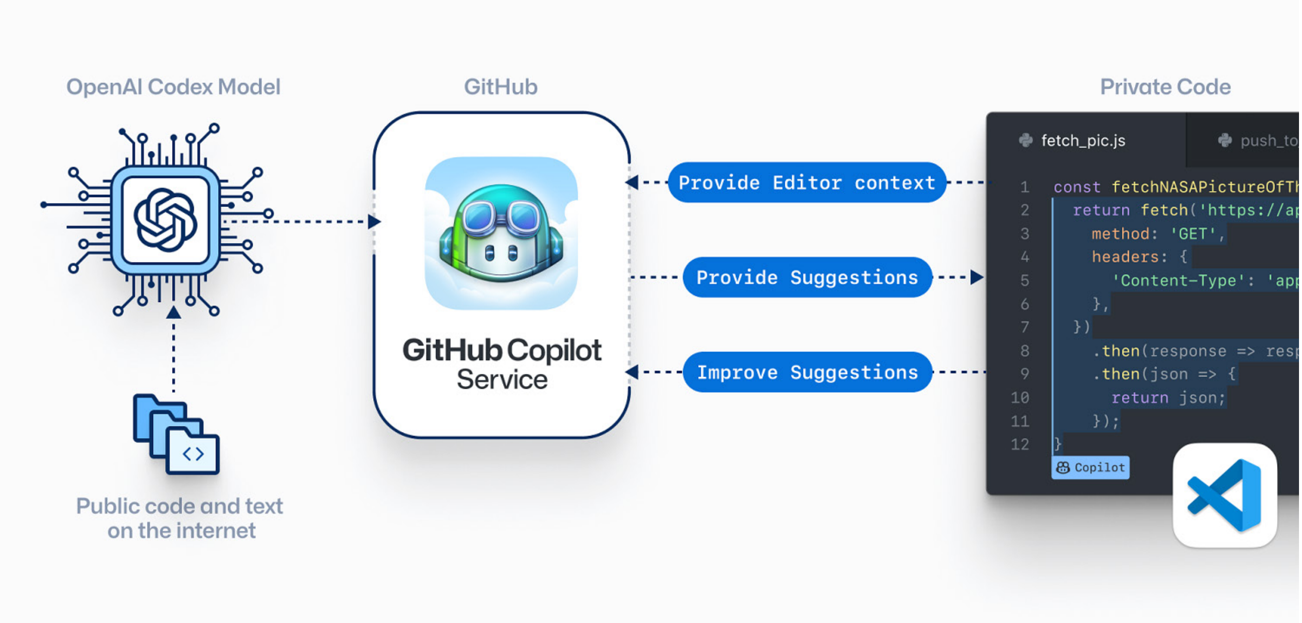


Figure 3. The path from training data to generated artefacts

For illustrative purposes, we take one well known example to the open source communities - Github Copilot, which is a service offered by Microsoft that essentially synthesises code snippets or full programs from natural language prompts.



```
const fetchNASAPictureOfTh
  return fetch('https://ap
    method: 'GET',
    headers: {
      'Content-Type': 'app
    },
  })
    .then(response => resp
    .then(json => {
      return json;
    });
}
```

Figure 4. GitHub Copilot [1]

One of the main questions that has been raised when it comes to this particular service was whether the use of publicly available open source repositories hosted on Github was compatible with the licence conditions under which this code had been released.

Basically, we have publicly available code and text on the Internet. These code repositories have been scraped and used as training data for the model. The model then produces code, which, in the words of Microsoft itself, should be considered private code. So, this transition from code to data to code raises the question how to reason about these transitions in copyright terms. What does this transformation from one phase into the other mean in legal terms?

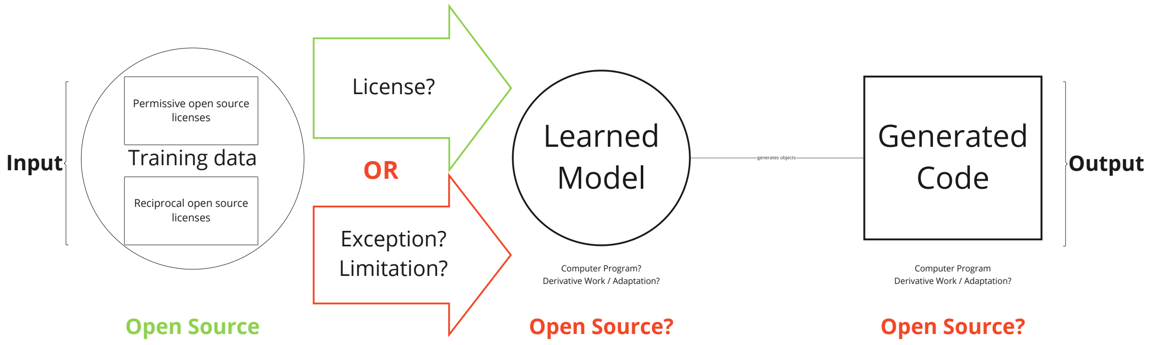


Figure 5. The transformation of code into data into code

[1] https://copilot.github.com/

The simple answer to that question is that it may not even matter, provided Microsoft could rely on a statutory exception or limitation. Typically, any act of reproducing a piece of code that is not purely functional and is protected by copyright requires the authorisation of the author of the computer program. However, if the law itself provides for an exception or limitation to this exclusive right of the author, then it no longer matters what the author says, because the legislator has balanced the interests of the author with those of the society so as to allow third parties to use this code without the need of authorisation.

Clearly, at least some form of reproduction of original works is required in the preparation and curation of data, so this is an act that would typically require an authorisation from the rights holder. That is, unless a statutory exception exists. In the EU, we have such an exception, also known as the text and data mining exception under Article 3 and 4 of the 2019 Copyright in the Digital Single Market (CDSM) Directive.

Even if we can overcome the requirement to comply with the licence conditions, which will not even matter if the statutory exception applies, the question still remains whether this creates a stable enough lawful basis for training models.

What Type of Activities Does the EU Text and Data Mining Exception Cover?

The CDSM Directive, which governs the text and data mining exception, provides that reproductions and extractions may only be retained for so long as necessary for the purposes of text and data mining.[2] The definition of 'text and data mining' (TDM) in Article (2) CDSM Directive is a catch-all provision that applies to any automated analytical technique aimed at analysing text and data in digital form in order to generate information which includes but is not limited to patterns, trends and correlations.[3]

The purpose of text and data mining is therefore to generate (new) information, such as patterns, trends, correlations. It is not entirely clear whether this generated information is limited to just the information extracted from the data (e.g., the trained model), or captures also new information generated on the basis of the text and data mining (e.g., generated code). An extensive interpretation of the TDM definition captures all generated information, regardless of whether it is, for example, patterns or correlations, or new objects generated on the basis of these patterns and correlations. A narrower construction of the text limits the exception to the information generated on the basis of the training data and does not cover any further use of the produced model. It is a principle of law that exceptions must be narrowly construed to avoid undermining the general rule.[4] We therefore argue that *the exception should be understood as covering only the process of generating a (working) model*.

[2] Directive (EU) 2019/790 of the European Parliament and of the Council of 17 April 2019 on copyright and related rights in the Digital Single Market and amending Directives 96/9/EC and 2001/29/EC, Art 4, paragraph 2.

[3] Directive (EU) 2019/790 of the European Parliament and of the Council of 17 April 2019 on copyright and related rights in the Digital Single Market and amending Directives 96/9/EC and 2001/29/EC, Art 2(2).

[4] Exceptio est strictissimae applicationis.

This understanding of the definition is also aligned with the temporal limitation ('as long as is necessary') to the exception. This limitation makes sense in the initial stages of preparing the data. It effectively obliges entities that apply text and data mining to delete extractions or reproductions from lawfully accessed works as long as this first phase of preparing the data has been completed. But can it also apply to cases where the model has memorised verbatim pieces of the training data?

What is the Temporal Limitation of the EU Text and Data Mining Exception and Does It Cover Memorised Data?

Recent work in deep learning has established a symbiotic relationship between memorisation and generalisation.[5] Simply put, we want a model to be able to generalise well on unseen data as opposed to overfitting or unintentionally memorising its training data.

From a copyright point of view, memorisation is simply a reproduction of (part of) a work and as such is an act that requires either authorisation from the rights holder, or a statutory exception. However, if we accept that there is indeed a symbiotic relationship between some forms of memorisation and generalisation, then we can argue that memorisation is essential to the purpose of text and data mining, which is to produce a working model that can generate information. This means that the exception also covers cases where memorisation of certain pieces of information from the training data is necessary for the trained model to perform well, ie, to generalise on unseen data. In this case, the extraction and reproduction, whether temporary (in the training process) or permanent (as correlations between data points in the trained model), would still be necessary for the purpose of text and data mining, i.e., to generate new information, and would therefore fall under the material scope of the exception.

That being said, one question remains: does the exception apply to outputs that may reproduce memorised details about certain data points, provided such memorisation is necessary for the model to perform well? To us, this is one possible case where the TDM exception reaches 'beyond' the trained model to apply to the output generated by a model. This implies, of course, that courts will have to employ quantitative assessment and rely heavily on expert testimony to be able to tell if the case concerns exempted unintended memorisation and not just poorly performing or overfitting models.

Is Numerical Data Copyrightable?

The preparation of data involves a process of decomposing the input data as tokens. For text, a token can be a word, part of a word (subword), or indeed a character. These tokens are then represented numerically as semantically useful units in the form of vectors which represent the meaning of a token relative to other tokens.

[5] Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, pages 123–132, 2021; Vitaly Feldman. Does learning require memorization. A short tale about a long tail. CoRR, abs/1906.05271, 2019; Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems, 33: 2881–2891, 2020.*

Essentially, this process entails transformation of copyright protected works, for example, a short story, into numerical values. Does copyright extend to this transformed subject matter? This raises the question whether and when the copyright protected work becomes so decomposed that it can no longer be reconstructed back or perceived as a 'work', and therefore becomes an independent element detached from the original protected work. In this process of tokenisation, the work is literally decomposed into building blocks by splitting words, phrase, sentences, and paragraphs. If we merely extract the meaning of and relations between these microscopic building blocks of the language that expressed the work in an objective form, then this can hardly be seen as reproduction of the work. The numerical representation of (parts of) a work into a collection of semantically useful units is also not an adaptation of the work because the right of adaptation anticipates the creation of a new work based on the original work. Tokens are mapped to vectors, which are then passed into neural networks, so copyright eligible subject matter is nowhere to be found in the process, ie, there is no new 'work'. In light of this analysis, we argue that the extracted information decomposed as tokens and represented numerically falls outside the remit of copyright. In other words, numerical data are not copyrightable.

One possible objection to this conclusion is that copyright law protects both source code and object code, which is, in a way, a numerical representation of the source code. The main argument is that the object code is just another representation of the source code and, as such, the two should be treated as one.[6] This is a clearly established position in both law and practice. However, we argue that the case with the representation of a work as tokens and vectors is different from compiled object code for the following reasons:

- Object code is an executable form of source code that has been optimised by the compiler and linker in a way that allows it to run on a particular computer architecture. The tokenization of a sentence is merely decomposing the building elements of the language of that sentence into smaller tokens. The numerical representation of these tokens as vectors reflects the meaning of a token relative to other tokens but is not itself a different form of the work.
- Object code is a computational artefact that represents the whole work. It is the complete translation of a program expressed in source code into an executable form. In contrast, the numerical representation of a part of, say, a literary work, provides information about the structural elements of the work's language and structure, but it tells us nothing about its aesthetic or artistic merits. To the untrained eye, reading this numerical representation would be "as useful as drinking a glass of diesel oil instead of pouring it into the tank of his vehicle".[7]
- Object code represents the complete translation of source code into an executable binary. In contrast, the vector representation of tokens relative to other tokens is merely an intermediary step in a long technical process. It does not represent the complete work and the complete work is not used 'as such' in the training process.

[6] Article 10 TRIPs.

[7] Opinion of Advocate General Szpunar delivered on 10 September 2019 Nederlands Uitgeversverbond and Groep Algemene Uitgevers v Tom Kabinet Internet BV and Others Request for a preliminary ruling from the Rechtbank Den Haag Reference for a preliminary ruling — Harmonisation of certain aspects of copyright and related rights in the information society — Directive 2001/29/EC — Article 3(1) — Right of communication to the public — Making available — Article 4 — Distribution right — Exhaustion — Electronic books (e-books) — Virtual market for 'second-hand' e-books Case C-263/18 (ECJ) [57].

Do I Need to Comply with the Licence Conditions When Using Open Source Code as Training Data?

In the EU, if a company relies on the TDM exception, any acts of temporary or permanent reproduction in the preparatory phase where data are collected, cleaned etc., will be covered by this exception. This means that, as a matter of law, the licence conditions will simply not apply. In any case, for many open source licences, the licence conditions are triggered only upon distribution, so the issue should not really arise in the preparatory phase.

Do I Need to Comply with the Licence Conditions When My Model Is Used?

If your model reproduces verbatim code that is identical to code released under an open source licence and you cannot rely on the TDM exception, based on the interpretation above, then, at the very least, you would have to comply with the notice and attribution requirements in the case of permissive licences, and the additional obligations in the case of reciprocal licences. Importantly, this only applies to the situation where the reproduced code is completely identical to existing code and is not of a purely functional nature (e.g., programmatic interfaces) and so it can qualify as an original work.

Hybridity of AI

We can distinguish three types of property layers in AI where exclusive rights might subsist in some of the subject matter.

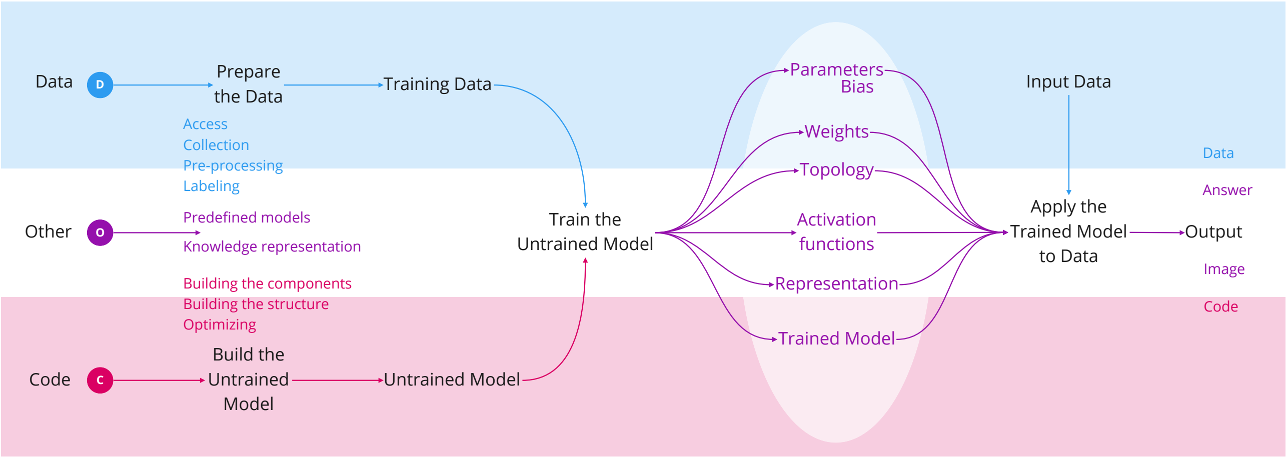


Figure 6. The hybrid nature of AI

The top layer concerns the data, but it includes other artefacts that are close to data, such as the weights, which we cannot unequivocally qualify as data but that still resembles data.

The bottom layer concerns code where copyright clearly subsists in at least the Jupyter notebooks, for example, and the training scripts. The status of the model, however, is far from clear under copyright law. Technically speaking, the model is the collection of weights and biases that are the learnable parameters of some machine learning models. We argue that a machine learning model is not copyright eligible subject matter because it is an algorithm that has been optimised to solve a general or more specific task, but is abstract in nature, is not expressed in a human-readable form, is not the result of creative choices, and cannot be attributed clearly to a human author. In the EU, one may be able to obtain sui generis database protection over the model, provided the qualitative and/or quantitative investment criteria are met.[8]

The middle layer concerns other categories of elements, such as annotations and knowledge representations. This layer sits in between humans and the machine and can have some artefacts that attract copyright protection. For example, ontology-based knowledge representations may be protected by copyright. Similarly, data annotations, especially where they contain expressive descriptions that pass the threshold of originality, could also obtain copyright protection.

The above picture shows the hybrid nature of machine learning where data, code and other computational artefacts co-exist in one single container, that is, the machine learning pipeline. How does intellectual property law treat such hybrid intellectual property?

AI as Hybrid Intellectual Property

In the context of the EU, the Court of Justice of the European Union provides some hints about the approach to hybrid intellectual property with an example from the domain of video games. In Nintendo v Pc Box,[9] a case about technological protection measures for video games, the Court provided a very interesting reasoning on the intellectual property nature of video games, acknowledging that they “constitute complex matter comprising not only a computer program but also graphic and sound elements, which, although encrypted in computer language, have a unique creative value which cannot be reduced to that encryption”.[10] The Court went on to say that “in so far as the parts of a videogame, in this case, the graphic and sound elements, are part of its originality, they are protected, together with the entire work, by copyright in the context of the system established by Directive 2001/29”.

[8] Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, Art 7.

[9] Case C-355/12, Nintendo v Pc Box, ECLI:EU:C:2014:25.

[10] Ibid, para 23

In practical terms, this decision is believed to solidify the distinction between closed subject matter systems and open subject matter systems in European copyright law.

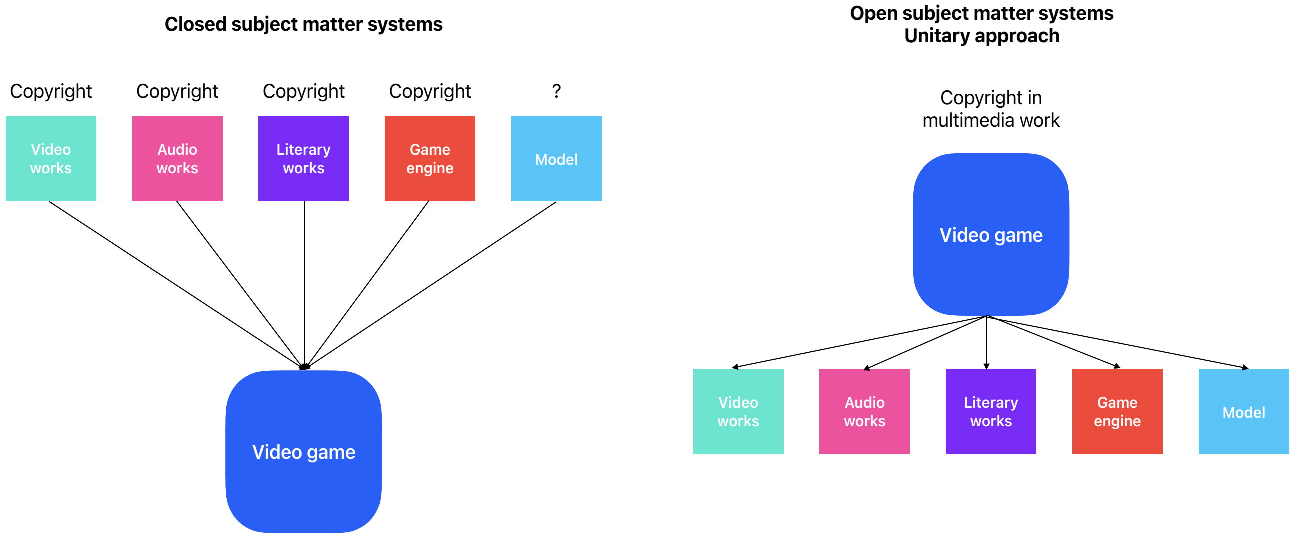


Figure 7. Closed and open subject matter system in European copyright law

We call the open system of IP subject matter a 'catch-all' system. So long as something is considered 'original', as in 'the author's own intellectual creation' (the EU standard for originality), it does not matter what it is as subject matter. Protection attaches automatically. In the closed subject matter system, we examine each work individually. Thus, for example, a video game would probably qualify as software under the special legal regime for computer software and not as a multimedia work under the closed subject matter regime. In the open system, that same game would likely qualify it as an independent work under the general rules of the Information Society Directive.

What is the practical difference between the two approaches? In our view, individual elements are assessed anyway to reach a conclusion whether the hybrid object is original or not. However, in the unitary approach of the open subject matter system, it is possible to also catch unprotected subject matter. Think of the hybrid work as a strong magnet that attracts subject matter.

Obviously, AI is different from video games. However, if we take the approach suggested by the court that the fact that the subject matter may not (always) be human-readable does not deprive it from its originality, then it is at least plausible to argue that AI could be seen as hybrid subject matter, as a matter of European copyright law at least. We are not arguing that this is good copyright policy and do not find copyright to be a good legal vehicle to capture the complexity of AI.

Then, of course, we must exercise caution in drawing general conclusions from just one case. Importantly, the Nintendo case is not just a case about video games, but more specifically about technological protection measures. In this case, the court dealt with the question whether the video games at hand were to be explored under the general regime of the Information Society Directive or under the lex specialis regime of the Computer Programs Directive. The court took the view that the Information Society Directive applies when dealing with hybrid media, and the Computer Programs Directive applies only in cases of 'pure' software.

In the context of our discussion, this raises an even more interesting question of whether we should consider AI, or at least some of the elements in the pipeline, as software or as generic copyright-eligible subject matter. If we take models to be merely intermediary elements in a more complex technical process which usually results in functioning applications, then, under the unitary approach, it is not impossible to consider that the copyright in the application layer may extend beyond and capture also the underlying models. That is, unless we argue that they are merely mathematical algorithms that are abstract in nature, as explained above.

Ultimately, we are faced with two questions:

- If we follow the unitary approach of the open subject matter system, could and should AI be licensed as a single object? How would a hypothetical reciprocal licence work in practice (e.g., what amounts to the Corresponding Source[11])?
- If we follow the closed subject matter system, which elements in the pipeline could and should attract copyright protection? How to transact with subject matter that is not covered by copyright and, more importantly, how to ensure licence compatibility across different subject matter?

[11] See GPLv3, §1. Source Code where "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work".

Building Blocks of Open Source AI

Some of the key questions when building open source AI concern the interaction between humans and machines. Unlike source code, which is human-readable, many of the core artefacts in machine learning do not exist in a human-readable source.

This implies that the effects strived for by the use of open source software cannot be achieved merely with the tools of copyright law because copyright subsists in only a small subset of the components that make up machine learning. Furthermore, unlike copyright, we do not have any international harmonisation of most of the rights that could subsist in AI, and we also do not have a single sui generis right. Open source licences are built around the underlying assumption that there exists source code to which copyright is attached, and it is precisely this copyright that is subject to licensing. Whether you relinquish your rights by dedicating a work to the public domain, or allow downstream recipients to use your work unconditionally or conditionally, all of these acts are, in essence, exercising your copyright in the code. Since we don't have anything remotely close to this kind of harmonised approach for AI, we suggest to focus on the effects of open source development and practices that we would like to replicate for AI.

Whether we look at AI as a hybrid type of an object or merely as a collection of computational artefacts, some of which may be copyright eligible, we should pay attention to the effects the components of AI create and how those components interact with each other. An effects-based, as opposed to rights-based, approach should be taken as the basis for defining open source AI. We suggest that the principles of transparency, enablement and reproducibility are taken as the key concepts to unlock the beneficial effects for open source for AI. Through these principles, we can identify how the components generated or used in the AI pipeline should feed into the development cycle of open AI enabling similar type of dynamism as we can see taking place in open source software communities.

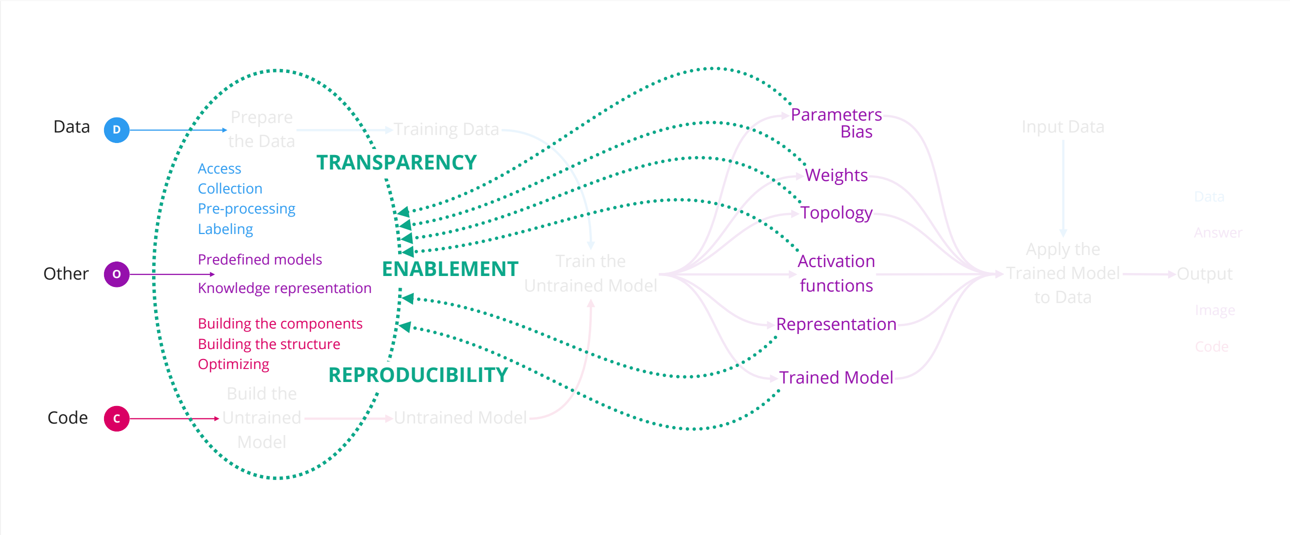


Figure 8. The three principles of Open Source AI in European copyright law

In our view, the above mentioned 3 essential principles – transparency, enablement and reproducibility – should lie at the heart of any definition of open source AI.

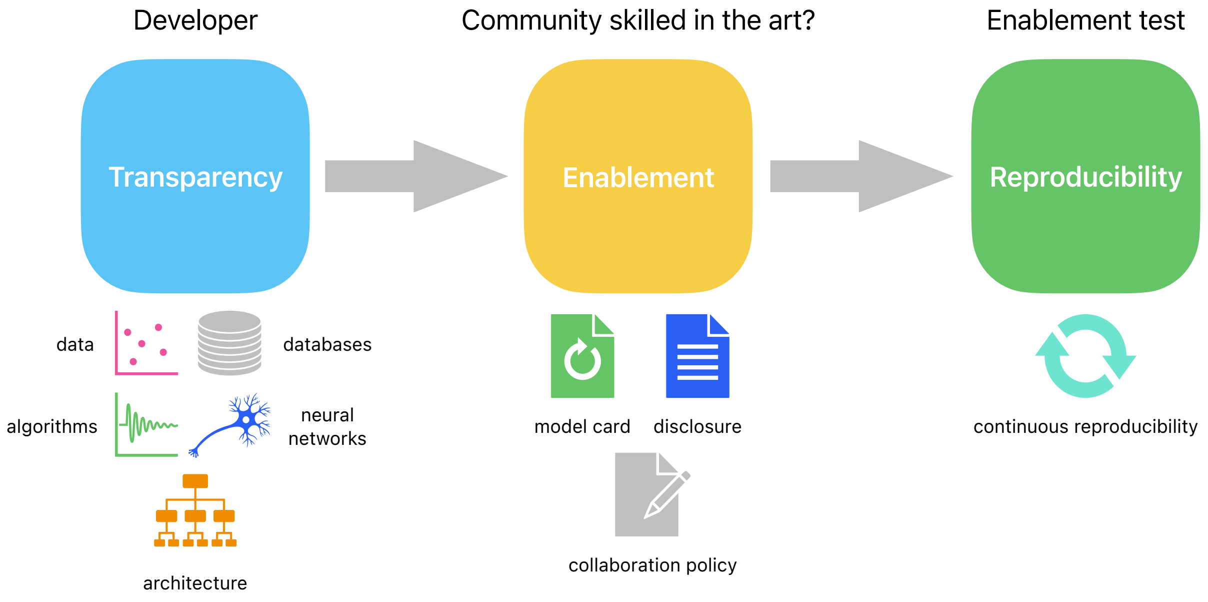


Figure 9. The path from transparency to reproducibility

- Transparency is understood as disclosure of details about the composition of training data sets, details about the data structures, architecture and algorithms, access to neural network weights etc. This is transparency in the technical sense of the word, implying such a level of disclosure that is conducive to enable the community around the project to rebuild the model, if it so wishes.

- Enablement is a principle akin to what we know from patent law as sufficiency of disclosure, that is, disclosing an invention in sufficient detail so that the person skilled in the art could carry out that claimed invention. Obviously, open source communities rarely have similar institutional capacity as a patent office to implement anything remotely similar. However, enablement, understood as disclosure of sufficient details about the building of a model to enable anyone to rebuild the model, at least theoretically and provided they have the computational resources, would allow a predictable standard for when an ML model is open source and when it is not. Obviously, enablement would depend on technical criteria developed by the communities around projects for following the principles of open source.

- A statement of enablement without a way of verifying the plausibility of this statement would mean little. We propose reproducibility as the third principle of open source AI. We are not talking about reproducibility in general, but rather of something more akin to reproducible builds, understood as a set of software development practices that create an independently-verifiable path from source to binary code. Of course, this proposition raises an additional concern about the feasibility of re-training and updating base models (not least for environmental concerns), and about the possibilities to avoid it. And it also raises the question of what should count as convincing evidence of reproducibility in the eyes of the community. In any case, if open source AI is to replicate the success of open source software, then downstream users should be able to rely on some form of community guarantees that the model is and will remain open source.

A Working Definition for Open Source AI

Based on the observations in the previous sections and our strive for effects-based definition, we propose the following working definition for Open Source AI:

Open Source AI, in the presently dominant form of machine learning, doesn't just mean access to models, weights, biases, algorithms or training and testing data. The communities around Open Source AI projects should be enabled to use, study, modify and share their modifications. Open Source AI must comply with at least the following basic criteria:

- Transparency: disclosure of details about the composition of training data sets, details about the data structures, architecture and algorithms, access to neural network weights etc.

- Enablement: disclosure of sufficient details about the building of a model to enable anyone to rebuild the model, provided they have access to the required computational resources, as identified by the community developing the AI.

- Reproducibility: development practices that create an independently-verifiable path from the training data to model inference.

The next step in expanding this definition would be to map these three key requirements to the 10 principles of the Open Source Definition (OSD) and to assess how they could be used in creating the effects mimicking weak and strong copyleft effects of open source software. The purpose of this exercise would be to identify which elements of the OSD can be kept intact, which need to be adapted, and which need to be discarded.

We encourage policy makers and other stakeholders to engage with the Open Source Initiative (OSI) in creating the Open Source AI Definition. Under the stewardship of the OSI, Open Source AI could become a dominant approach to machine learning, one that will allow communities, individuals and companies to reap the benefits of collaborative development, testing and deployment.

ZOOOM

www.zooom4u.eu

Funded by the European Union

Grant agreement No. 101070077
Duration: 10/2022 – 09/2024
Funded by the European Union. Views and opinions expressed are however those of the ZOOOM Consortium and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them.