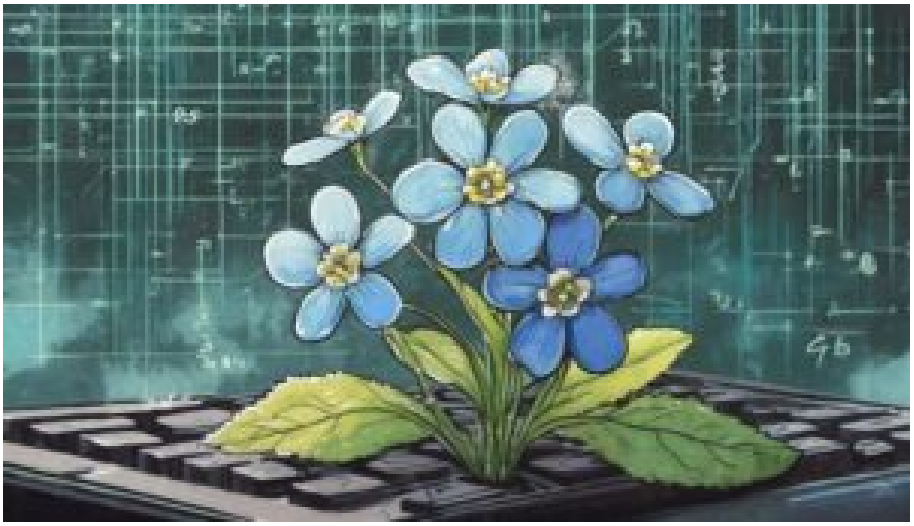
Kluwer Copyright Blog

Memorisation in generative models and EU copyright law: an interdisciplinary view

Ivo Emanuilov, Thomas Margoni (KU Leuven Centre for IT & IP Law) · Tuesday, March 26th, 2024



Art illustration generated using the Adobe Firefly Image 2 model with the following prompt: "Draw an art illustration with the forget-me-not flower as an illustration of memorisation in machine learning with matrix calculations in the background"

Large language models' (LLMs) greatest strength may also be their greatest weakness: their learning is so advanced that sometimes, just like humans, they memorise. This is not surprising, of course, because computers are really good at mainly two things: storing and analysing data. There is now empirical evidence that deep learning models are prone to memorising (i.e., storing) fragments of their training data. Just like the human brain needs to memorise fragments of information to learn, so do LLMs. And when they reproduce verbatim these fragments, this may be a ground for copyright infringement.

Enter the Transformer

The transformer architecture (as in Generative Pre-trained *Transformer*, GPT) enabled many new applications but, arguably, the most impressive one remains synthetic content generation, such as text, images and video. The key to the success of transformer technology is the ability to *generalise*, that is, to operate correctly on new and unseen data. Traditionally, the ability to generalise is at odds with memorization. Memorization is much like in humans: if you memorize the answers to an exam, you'll probably perform well if the exam's questions are identical to those you practised. But the more you are asked to apply that knowledge to a new scenario the more your performance drastically diminishes. You have failed to *understand* what you learned; you only memorized it. Transformers, from this point of view, work not too differently: they aim at understanding (generalising), but they may memorise in certain situations.

It Is important to clarify that, from a technical point of view, transformer-based models encode words as groups of characters (i.e., tokens) numerically represented as vectors (i.e., embeddings). The models use neural networks to maximise the probability of every possible next token in a sequence, resulting in a distribution over a vocabulary which consists of all words. Each input token is mapped to a probability distribution over the output tokens, that is, the following characters. This is how transformers "understand" (or generalise, or abstract from) their training data. The models, however, do not memorise the syntax, semantics, or pragmatics of the training data (e.g., a book, poem, or software code). They instead learn patterns and derive rules to generate syntactically, semantically, and pragmatically coherent text. Even if the 'source code' of a large language model could be made available, it would be virtually impossible to revert back to the training data. The book is not present in the trained model. However, the model could not have been developed without the book.

The many faces of memorisation

One common fault in non-technical literature is the frequent belief that all machine learning algorithms behave in the same way. There are algorithms that create models which explicitly encode their training data, i.e., memorisation is an intended feature of the algorithm. These are, for instance, the ?-nearest neighbour classification algorithm (KNN), which is basically a description of the dataset, or the support vector machines (SVM), which include points from the dataset as 'support vectors'.

Similarly, non-technical literature rarely distinguishes between *overfitting* (too much training on the same dataset which leads to poor generalisation and enhanced memorisation) and forms of *unintended memorisation* which instead may be essential for the accuracy of the model.

As a matter of fact, recent research shows that memorisation in transformer technology is not always the result of a fault in the training process. Take the case of the memorisation of rare details about the training data, as argued by Feldman. His hypothesis draws on the long-tailed nature of data distributions and purports that memorisation of useless examples and the ensuing generalisation gap is necessary to achieve close-to-optimal generalisation error. This happens when the training data distribution is long-tailed, that is, when rare and non-typical instances make up a large portion of the training dataset. In long-tailed data distributions, useful examples, which improve the generalisation error, can be statistically indistinguishable from useless examples, which can be outliers or mislabelled examples. Let's illustrate this with the example of birds in a collection of images. There may be thousands of different types or species of birds, and some

subgroups may look very different because of different levels of magnification, or different body parts, or backgrounds that are highlighted in the image. If the images are categorised simply as ‘birds’ without distinguishing between specific subgroups, and if the learning algorithm hasn’t encountered certain representatives of a subgroup within the dataset, it might struggle to make accurate predictions for that subgroup due to their differences. Since there are many different subpopulations, some of them may have a low frequency in the data distribution (e.g., 1 in). For a subgroup of birds, it may be that we would only observe one example in the entire training data set. However, one may also be the number of outliers our algorithm would observe. The algorithm wouldn’t be able to distinguish between something genuinely rare and an outlier that doesn’t represent the majority of the data. Similarly, in areas where there is a low confidence, the algorithm would not be able to tell a “noisy” example from a correctly labelled one. If most of the data follows a pattern where some types of birds are very rare and others are more common, these rare occurrences can actually make up a significant portion of the entire dataset. This imbalance in the data can make it challenging for the algorithm to learn effectively from it.

Long-tailed data distributions are typical in many critical machine learning applications from face recognition, to age classification and medical imaging tasks.

Table 1 Different forms of memorisation

MEMORISATION TYPE	CAUSE	EFFECT
Intended memorisation (algorithm design)		
Memory-based learning (eg, k-Nearest Neighbours (kNN) algorithm)	The algorithm replaces model creation by memorising the training data set and then use this data to make predictions.	Encodes part of the data set. Highly accurate predictions but poor performance on large datasets or high dimensionality.
Support Vector Machines (SVM)	The algorithm uses a subset of training points from the dataset in the decision function (support vectors)	Encodes part of the dataset as support vectors. Effective in high dimensional spaces.
Unintended memorisation that does not improve accuracy		
Overfitting	Most often the result of overtraining the model, i.e., training for too many times on a single sample set of data.	Contains more parameters than can be justified by the data (overparametrised). Model performs very well on the training data set or on data sets very similar to the training one but will perform very poorly when confronted with unexpected environments.
Eidetic memorisation	Trained neural networks encode out-of-distribution training data, ie, training data that is irrelevant to the learning task and unhelpful to improving model accuracy.	Encodes parts of the dataset. Model does not perform better.
Unintended memorisation necessary for generalization		
Long-tailed memorisation	Memorisation of useless examples and the ensuing generalisation gap is necessary to achieve close-to-optimal generalization error. Occurs in long-tail data distributions a significant fraction of which are made up of rare and atypical instances.	Encodes parts of the datasets (label memorisation). Memorisation can be beneficial to and, in fact, needed for generalisation when the data distribution has a long-tail nature.

The Text and Data Mining (TDM) exceptions and the generation of synthetic content

The provisional compromise text of the AI Act proposal seems to clarify beyond any doubt (if there was any) that CDSMD's TDM exceptions apply to the development and training of generative models. Therefore, all copies made in the process of creating LLMs are excused within the limits of Art. 3 and 4 CDSMD. In the CDSMD there seems to be a sort of implicit assumption that these copies will happen in the preparation phase and not be present in the model (e.g. Rec. 8-9). In other words, the issue of memorization was not directly addressed in the CDSMD. Nevertheless, the generous structure of Arts. 2 – 4 CDSMD is arguably sufficiently broad to also cover permanent copies eventually present in the model, an interpretation that would excuse all forms of memorization. It should be noted, of course, that a model containing copyright relevant copies of the training dataset cannot be distributed or communicated to the public, since Art. 3 and 4 only excuse reproductions (and in the case of Art. 4 some adaptations).

Regarding the *output* of the generative AI application and whether copyright-relevant copies eventually present there are also covered by Art. 3 and 4 the situation is less clear. Nevertheless, even if those copies could be seen as separate and independent from the ensuing acts of communication to the public, this solution would be quite ephemeral at the practical level. In fact, those copies could not be further communicated to the public due to the very same reasons pointed out above (Arts. 3 and 4 only excuse reproductions, not communications to the public). The necessary conclusion is that if the model generates outputs (e.g., an answer) that may qualify as a copy in part of the training material, these outputs cannot be communicated to the public without infringing on copyright.

A situation where the generative AI application does not communicate *its model* but only the generated outputs (e.g., answers) is perfectly plausible, and in fact makes up most of the current commercial AI offerings. However, an AI application that does not communicate *its outputs* to the public is simply hard to image: it would be like having your AI app and not be able to use it. Of course, it is possible to have the outputs of the model not directly communicated to the public but used as an intermediary input for other technical processes. Current developments seem to be in the direction of applying downstream filters that remove from the AI outputs the portions that could represent a copy (in part) of protected training material. This filtering could naturally be done horizontally, or only in those jurisdictions where the act could be considered as infringing. In this sense, the deployment of generative AI solutions would likely include elements of copyright content moderation.

Should all forms of memorisation be treated the same?

From an EU copyright point of view, memorisation is simply a reproduction of (part of) a work. When this reproduction triggers Art. 2 InfoSoc Directive it requires an authorisation, either voluntary or statutory. However, if we accept that there is indeed a symbiotic relationship between *some* forms of memorisation and generalisation (or less technically, learning), then we could argue that this second type of memorisation is necessary for improved (machine) learning. In contrast, overfitting and eidetic memorisation are not only not necessary for the purpose of abstraction in

transformer technology but they have a negative impact on the model's performance.

Whereas we showed that EU copyright law treats all these forms of memorization on the same level, there may be normative space to argue that they deserve a different treatment, particularly in a legal environment that regulates TDM and Generative AI on the same level. For instance, most of the litigation that is emerging in this area is predicated on an alleged degree of similarity between the generative AI *output* and the *input* works used as training material. When the similarity is sufficient to trigger a *prima facie* copyright claim it could be argued that the presence or absence of memorization may be a decisive factor in a finding of infringement.

If no memorization has taken place, the simple "learning" done by a machine should not be treated differently from the simple learning done by a human. On the other hand, if memorization was present "unintentionally" the lack of intention could warrant some mitigating consequence to a finding of infringement, illustratively, by way of reducing or even excluding monetary damages in favour of injunctive relief (perhaps combined with an obligation to mend the infringing situation once notified, similarly to Art. 14 e-Commerce Directive, now Article 6 of the Digital Services Act.). Finally, situations where memorisation was intended or negligently allowed could be treated as normal situations of copyright infringement.

Naturally, the only way to prove memorisation would be to have access to the model, its source code, its parameters, and training data. This could become an area where traditional copyright rules (e.g., infringement proceedings) applied to AI systems achieve the accessory function of favouring more transparency in a field commonly criticised for its opacity or "black box" structure. Copyright 1, AI 0!

If you want to dig deeper into this discussion, please check out the preprint of our paper which provides an extensive discussion of memorisation through the lens of generative models for code. This research is funded by the European Union's Horizon Europe research and innovation programme under the 3Os and IP awareness raising for collaborative ecosystems (ZOOOM) project, grant agreement No 101070077.

This entry was posted on Tuesday, March 26th, 2024 at 10:30 am and is filed under Artificial Intelligence (AI), European Union
You can follow any responses to this entry through the Comments (RSS) feed. You can leave a response, or trackback from your own site.