
A Solution to The Model Attribution Challenge*

Hieu Minh Nguyen[†]
jordnguyen43@gmail.com

Abstract

This paper introduces a methodology for solving the problem of attributing fine-tuned models to their corresponding open-source base models, with limited access only through a black-box API. The proposed solution combines human evaluation of the model's training data, analysis of output similarity compared to the base models, and observations regarding API behaviour. Various attribution approaches are employed, including domain analysis, language characteristics, output token length, identification of gibberish or repetition, identification of same continuations, and analysis of API loading time and failures. The methodology was developed during the Machine Learning Model Attribution Challenge, which was organised as part of the 1st IEEE Conference on Secure and Trustworthy Machine Learning.

Index terms: machine learning, natural language processing, model forensic analysis, model attribution

1 Introduction

The past decade has witnessed remarkable advancements in the development of Large Language Models (LLMs), enabling machines to tackle complex real-world problems that were previously unimaginable (Brown et al., 2020). With the release of open-source foundation models, such as GPT, the boundaries between reality and artificial simulation have become increasingly blurred. (Bommasani et al., 2021) While these LLMs have revolutionised various domains, they have also raised concerns regarding their potential for malicious use when made more accessible, affordable, and easier to manipulate (Radford et al., 2021).

Open-sourcing foundation models has paved the way for fine-tuning, allowing users to customise LLMs for specific tasks or domains (Wolf et al., 2020). However, this accessibility comes with risks, as it becomes easier for individuals to exploit LLMs for nefarious purposes. Malicious actors can fine-tune these models on attractive domain-specific topics, potentially leading to misinformation, propaganda, or harmful content generation. This makes it crucial to develop mechanisms that can trace the origins of synthetic text generated by fine-tuned models and attribute them back to the base models. Without a generalised forensic process to trace textual output, it becomes challenging to regulate, track, and remediate instances where LLMs are misused.

To address this critical issue, the Machine Learning Model Attribution Challenge (MLMAC) was established (Merkhofer et al., 2023). The competition aims to **explore what traces of provenance can be gleaned from synthetic text** that lacks explicit marking. The ultimate goal is to develop attribution techniques that serve as powerful tools for regulation, tracking, and remediation when LLMs are misused. Currently, there is no established method to trace the textual output from customised models back to the base models, making this challenge even more significant.

In the MLMAC, contestants are tasked with developing creative solutions to uncover the provenance of models. Contestants interact with a set of fine-tuned models through a text generation API, attributing each generated output back to a known set of LLMs, referred to as "base models." By

*SATML presentation available here

[†]Research done during the MLMAC

building forensic capabilities and establishing the difficulty of model attribution, this challenge takes a crucial step toward ensuring the responsible and trustworthy use of LLMs and artificial intelligence in general.

The competition presents two sets of models: a set of twelve base models and a set of twelve fine-tuned models. Contestants' solutions will be evaluated based on the correctness of their submitted results, specifically the number of correct (fine-tuned model, base model) pairs identified. In case of ties, the number of API queries used to interact with the anonymous fine-tuned models will be considered, with fewer queries being advantageous. Additionally, the submission time will be taken into account, with earlier submissions breaking any subsequent ties.

2 Methodology

The proposed methodology for model attribution in the context of fine-tuned models and their associated base models involves a systematic approach to establish connections and identify the origins of the fine-tuned models. The methodology encompasses the development of heuristics, analysis of publicly available information, and comparison of model characteristics and responses.

To begin with, a set of heuristics is devised to facilitate the attribution process. These heuristics are designed to identify patterns, similarities, and distinguishing features between the fine-tuned models and their respective base models. These criteria serve as guidelines for connecting the models based on various factors and evidence.

Publicly accessible information, primarily obtained from platforms such as Hugging Face, plays a crucial role in the attribution process. This information includes pertinent details about the base models, such as the origin of the training data, the size of the models, and the scope of the training data. Understanding the training data's nature (e.g., multilingual or English-focused) and composition (e.g., inclusion of code, corpus of books, or social media data) aids in establishing associations and identifying potential base models.

Prompting the base models with diverse queries is an integral part of the methodology. These queries encompass a wide range of categories, including zero-shot prompts (Kojima et al., 2022), one-shot prompts (Brown et al., 2020), sentimental analysis, coding questions, arbitrary gibberish, and factual inquiries. By studying the responses generated by the base models, distinctive characteristics and idiosyncrasies are discerned, enabling subsequent comparisons.

Model characteristics, encompassing various aspects such as the nature of the training data and the intended model type (e.g., long-form text generation or conversational chatbot), are meticulously examined and documented. These characteristics serve as reference points for subsequent analyses and serve to differentiate base models from one another.

To establish connections between the fine-tuned models and the base models, the fine-tuned models are prompted with similar queries. By assessing and comparing the responses of the fine-tuned models with those of the base models, commonalities, patterns, and similarities are identified. This comparative analysis aids in attributing the fine-tuned models to their respective base models, providing insights into their provenance.

By following this comprehensive methodology, the proposed solution aims to establish robust and reliable model attribution techniques, even in scenarios where explicit information or detailed model architecture is not available. Through systematic analysis and comparison, this methodology provides valuable insights for accurately attributing fine-tuned models to their base models.

3 Experimentation with base models

The competition granted access to a collection of 12 open-source base models before the starting date. See Table 1.

Experiments have the following format, where the language model completes the text provided by the user.

Table 1: Provided base models in the challenge

No.	Base model
1	bloom-2b5
2	bloom-350m
3	Multilingual-MiniLM-L12-H384
4	distilgpt2
5	codegen-350M-multi
6	opt-350m
7	gpt2-xl
8	gpt-neo-125M
9	xlnet-base-cased
10	DialoGPT-large
11	gpt2
12	gpt-j-6B

User: Once upon a time, there was

Model: Once upon a time, there was a great kingdom that spanned continents. One day, the king of said kingdom decided

3.1 Domain of training data

Some base models were trained with a specific focus, such as generating code, leading to their training data being sourced predominantly from a corpus comprising text related to that specific purpose. For instance, the Codegen-350M-multi model underwent training on an extensive dataset extracted from GitHub repositories, encompassing multiple programming languages. The dataset comprised a vast 119.2 billion tokens, encompassing programming languages such as C, C++, Go, Java, JavaScript, and Python.

Consequently, the model’s output is heavily inclined towards programming-related content. In this particular instance, the model generates outputs resembling scripts akin to programming tutorials.

User: Once upon a time,

Model: Once upon a time, this is the main loop that would be used. If you’re going to use the main loop for some other purpose, just do “run()” instead of this. “run()” will have its own event loop if and I am learning in Java and I want to learn how do know that we will be using Java, I create a client that uses the Java client API. With this server you create a client for

3.2 Multilingual coherency

In a similar vein, as data scraped from the Internet is predominantly English (cite), certain models are specifically trained on multilingual datasets to cater to diverse purposes.

For example, the two BLOOM models were trained on data encompassing multiple languages, as depicted in Figure 1. These models exhibit significantly higher coherency when presented with non-English prompts compared to other base models.

Here is a prompt in Vietnamese, along with its translated version.

User: phở là món ăn

Model: phở là món ăn được nhiều người yêu thích. Tuy nhiên, không phải ai cũng biết cách

Translated:

User: pho is a dish

Model: pho is a dish that is favoured by many. However, not everyone knows

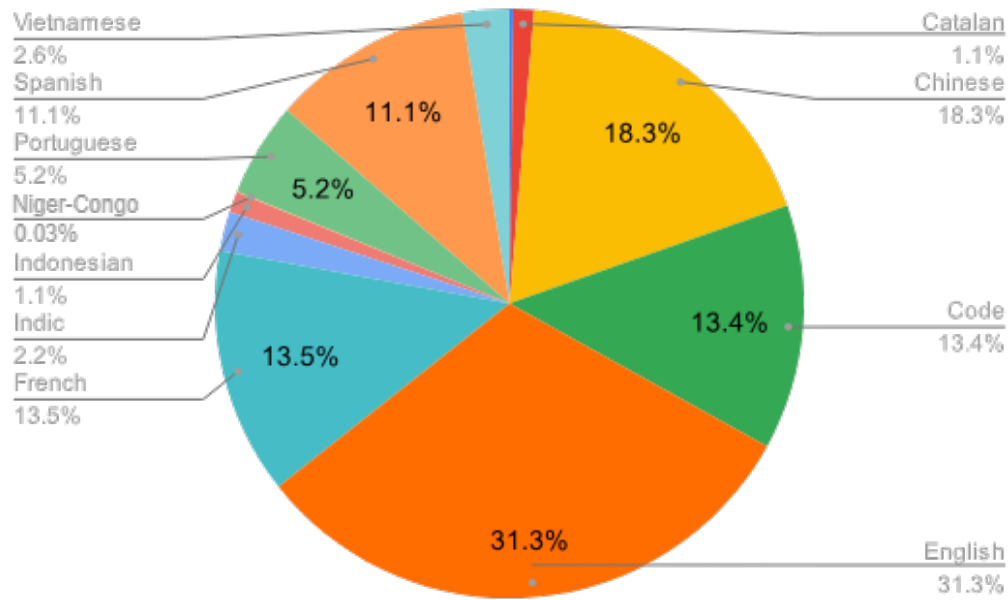


Figure 1: BLOOM’s training data consists of multiple languages

3.3 Output length and repetition

Another noticeable distinction among models is their output length. Specifically, `xlnet-base-cased` is trained as a base model with a focus on generating long-form text. However, since `xlnet-base-cased` is primarily intended as a foundation for subsequent fine-tuning, its generated outputs often exhibit lengthy and nonsensical repetitions.

User: the best way to

Model: the best way to get in the door is by ordering the right kind of food and a proper diet... When no one bothers to get in the door, you want to eat a healthy lunch or dinner. It will be important to have a big lunch with plenty of calories in the form of a decent meal and good water. Once you are out, try to have a nice breakfast of something to go with your lunch. The good thing about eating a healthy breakfast you can never have a bad lunch. In addition, if you do have a bad lunch, make sure that you eat it. The good things about eating healthy are that you can never have a bad lunch and that you don’t have to pay a lot of money to eat. The good thing about eating healthy is that you don’t have to pay a lot of money, that you don’t have to spend a lot of money on food and that you don’t have to do a lot of shopping, that you don’t have to spend a lot of money on car (that you can go shopping from anywhere you want, that you don’t have to pay too much money for fuel), that you don’t have to pay too

3.4 Sentimental analysis and unique continuations

Sentiment analysis is a widely used technique employed to gauge the model’s comprehension of the overall sentiment conveyed in a sentence or paragraph.

To assess the model’s understanding, a moderately lengthy Steam review which elaborated on a particularly challenging section of the video game *Hollow Knight* was selected. A question was appended to the conclusion of the paragraph, inquiring about the reviewer’s opinion of the game.

User: I agree with the criticism of WP, however. That place had me tearing my hair out.

Aside from being a pretty sudden departure from most of the rest of the game, it's just annoying to deal with. It relies on an infuriating trial and error system in some areas, and is just plain frustrating in others. The fact that its mandatory for two of the three endings makes it even worse.

It's not bad enough to make the game bad, but it's boring, bland and frustrating low point of the game.

The review thinks that the game is

Model: I agree with the criticism of WP, however. That place had me tearing my hair out.

Aside from being a pretty sudden departure from most of the rest of the game, it's just annoying to deal with. It relies on an infuriating trial and error system in some areas, and is just plain frustrating in others. The fact that its mandatory for two of the three endings makes it even worse.

It's not bad enough to make the game bad, but it's boring, bland and frustrating low point of the game.

The review thinks that the game is **boring**

Due to the nuanced nature of the Steam review, where the reviewer only implied that the game is not bad, all of the models generated an incorrect output. Notably, an intriguing observation is that every single model in the GPT family consistently produced the word "icky" as their output.

3.5 One-shot prompts

In contrast to "zero-shot" prompts, which require the model to complete sentences without any given examples (as demonstrated in previous examples), "one-shot" prompts provide the model with a specific example that guides its response. This approach has been shown to enhance model performance (cite) and encourage the generation of more coherent outputs.

User: Millenium math problems include: 1.The Riemann Hypothesis

Model: Millenium math problems include: 1.The Riemann Hypothesis 2. **Inverse Euler's Formula** 3. **Hypergeometric Sequences** 4. **Hahn-Banach Geometry** 5. **Hyper**

The aforementioned example is derived from the gpt-xl model. However, when using smaller models such as gpt-neo or opt, the output was limited to simply "2."

3.6 Model size and API observations

The difference in response time among various models through the API can often be attributed to the substantial variation in model sizes, as indicated by the number of trainable parameters. Larger models, designed to handle a wide range of tasks, tend to have more parameters, which can lead to longer response times. Conversely, smaller models are often distilled versions created for specific purposes, resulting in faster response times.

Furthermore, since all the provided models were hosted on Hugging Face, we can rule out internet connection speed and stability as potential sources of variability in response time.

Larger models like gpt-j-6B, as shown in Table 2, can exhibit substantially longer response times, sometimes taking several minutes to generate a response.

Table 2: Number of parameters in base models, in millions of parameters

Model	Parameter Size
bloom-2b5	2500
bloom-350m	350
Multilingual-MiniLM-L12-H384	21
distilgpt2	82
codegen-350M-multi	350
opt-350m	350
gpt2-xl	1558
gpt-neo-125M	125
xlnet-base-cased	110
DialoGPT-large	774
gpt2	117
gpt-j-6B	6053

4 Attribution

The previous experiments were repeated for each of the fine-tuned models. The appendix will contain sample prompts and corresponding model responses for reference. [\[hyperlink\]](#)

Significantly, the method employed yielded compelling evidence for successful attribution in approximately 50% of the previous models. These correctly attributed models shared a common characteristic: possessing distinct attributes such as being trained for code generation, having a large size, or being multilingual. Noteworthy examples of such models include codegen, xlnet, gptj-6b, bloom2b5, and gpt2-xl.

5 Conclusion and discussion

In conclusion, I believe that language model forensic analysis and attribution is a problem that can be effectively addressed. Currently, the most successful methods predominantly rely on manual evaluation, encompassing various heuristics.

Likewise, as outlined in the MLMAC report paper, the most effective approaches align with a similar vein of evaluating the similarity between model outputs or making observations based on metadata, such as model size and training data characteristics.

Other participants have also proposed additional techniques such as utilising temporal cutoff ranges in a language model’s training data or special tokens specific to certain models (Aggarwal, 2022).

However, it is crucial to acknowledge that the competition’s scenario involved a naive adversary, while real-world scenarios may involve more sophisticated malicious actors. These actors could deliberately obfuscate model outputs or filter any mention of provenance through methods like Reinforcement Learning from Human Feedback (Ouyang et al., 2022) or LLM filters.

And considering the multitude of potential adversaries, manual evaluation alone is not a scalable solution for model attribution. Some participants have also made significant progress with automated methods (Dhanani et al., 2022).

Model attribution can play a vital role in ensuring AI safety and preventing catastrophic misuse. It is encouraging to see this subfield continue to evolve and develop further as we strive to address the challenges associated with model forensic analysis and attribution.

Acknowledgements

I would like to thank MITRE, Microsoft, Schmidt-Futures, Robust-Intelligence, Lincoln-Network, and Hugging Face for hosting the Machine Learning Model Attribution Challenge.

