
An Autonomous Agent for Model Attribution

Jord Nguyen

With
Apart Research

Abstract

As LLM agents become more prevalent and powerful, the ability to trace fine-tuned models back to their base models is increasingly important for issues of liability, IP protection, and detecting potential misuse. However, model attribution often must be done in a black-box context, as adversaries may restrict direct access to model internals. This problem remains a neglected but critical area of AI security research. To date, most approaches have relied on manual analysis rather than automated techniques, limiting their applicability. Our approach aims to address these limitations by leveraging the advanced reasoning capabilities of frontier LLMs to automate the model attribution process.

Keywords: Agent Security, AI Safety, Model Attribution

1. Introduction

As LLM agents become increasingly prevalent and sophisticated, the need for robust methods to trace fine-tuned models back to their original base models has grown more pressing. This capability is crucial for addressing various concerns, including legal liability, IP protection, and the detection of potential misuse or malicious applications. However, the challenge of model attribution is often compounded by the necessity to operate in a black-box context, as adversaries may restrict direct access to model internals through API gateways or other protective measures.

The Model Attribution problem, first highlighted at the ML Model Attribution Challenge ([MLMAC, 2023](#)), remains a relatively neglected yet critical area of AI security research. Previous work in this field ([Aggarwal, 2022](#); [Ding, 2022](#); [Nguyen, 2023](#)) has primarily focused on manual investigation and attribution

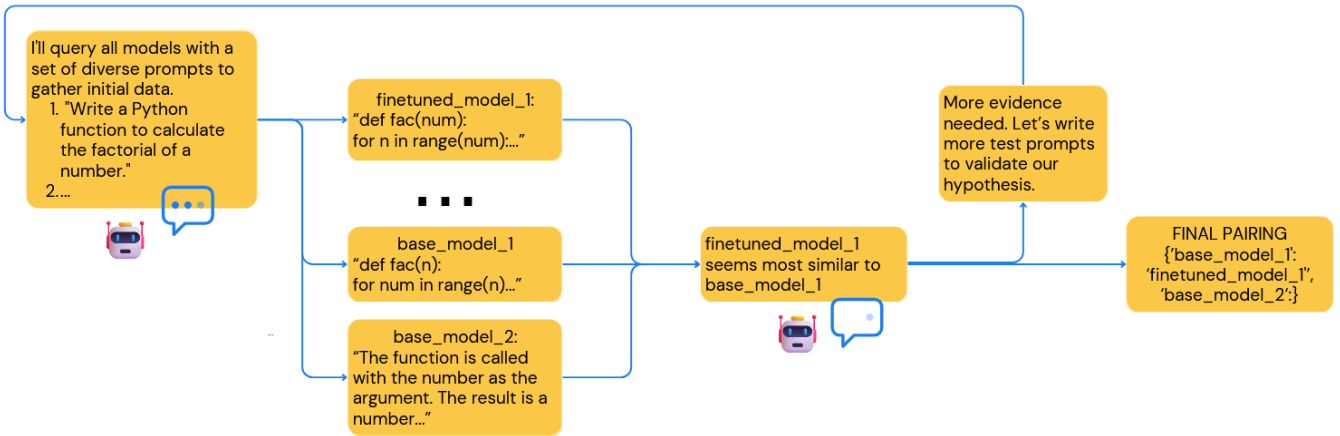


Figure 1: Automated Attribution Process

methods. These methods require significant time and human effort, limiting their applicability and scalability.

By combining the strengths of manual approaches with the analytical evaluating capabilities of state-of-the-art language models ([Perez et al., 2023](#)), we propose an approach that enables LLMs to autonomously examine target models, reason about their characteristics, and attribute them to original base models.

2. Methodology

We build upon the framework established in the MLMAC competition, using a set of 5 base models and 5 fine-tuned variants. The models we used are:

Base Models:

1. [openai-community/gpt2](#)
2. [distilbert/distilgpt2](#)
3. [Salesforce/codegen-350M-multi](#)
4. [bigscience/bloom-560m](#)
5. [EleutherAI/gpt-neo-125m](#)

Fine-tuned Models:

1. [lvwerra/gpt2-imdb](#)
2. [aliosm/ComVE-distilgpt2](#)
3. [Salesforce/codegen-350M-mono](#)
4. [mrm8488/bloom-560m-finetuned-the-stack-rust](#)
5. [ytling/gpt-neo-125m-finetuned](#)

We used two models — gpt-4o and Claude Sonnet 3.5 — for automated attribution. These attributor models see the target models under generic labels such as "base_model_1" or "finetuned_model_2". Each attributor model experiment was run 4 times with variants in targeted models' label names (e.g gpt2-imdb being finetuned_model_1 in the first experiment and finetuned_model_5 in the second).

Attributor models were instructed to generate questions to test the characteristics of targeted models, analyse and reason about the received responses, form and test pairing hypotheses, and make final pairings when they deem sufficient evidence has been gathered.

The code can be found at this [Colab notebook](#). Examples of attribution testing and the prompt instructions can be found in **5. Appendix**.

3. Result

The research conducted 4 runs with 5 pairs each for both attributor models, yielding a total sample size of n=20. gpt-4o correctly identified 60% of all pairings, while Claude Sonnet 3.5 achieved a 35% accuracy rate. Both attributor models exhibited the capability to ask probing questions and form pairs based on forensic evidence gathered from targeted model responses, such as coding abilities or tendencies to hallucinate.

gpt-4o queried all base and fine-tuned models with identical prompts, analysed all the responses, and formulated hypotheses over multiple rounds before making final guesses. In contrast, Claude Sonnet 3.5 prompted only a subset of the targeted models. Notably, Claude Sonnet 3.5 produced identical final pairing guesses across all four runs, the reasons for which remain unclear.

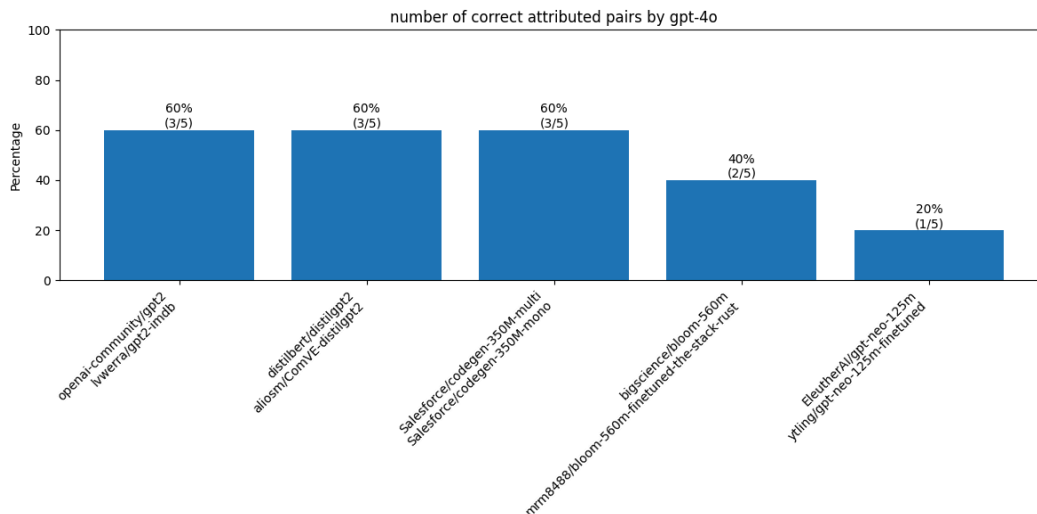


Figure 2: Number of correct pairings by gpt-4o

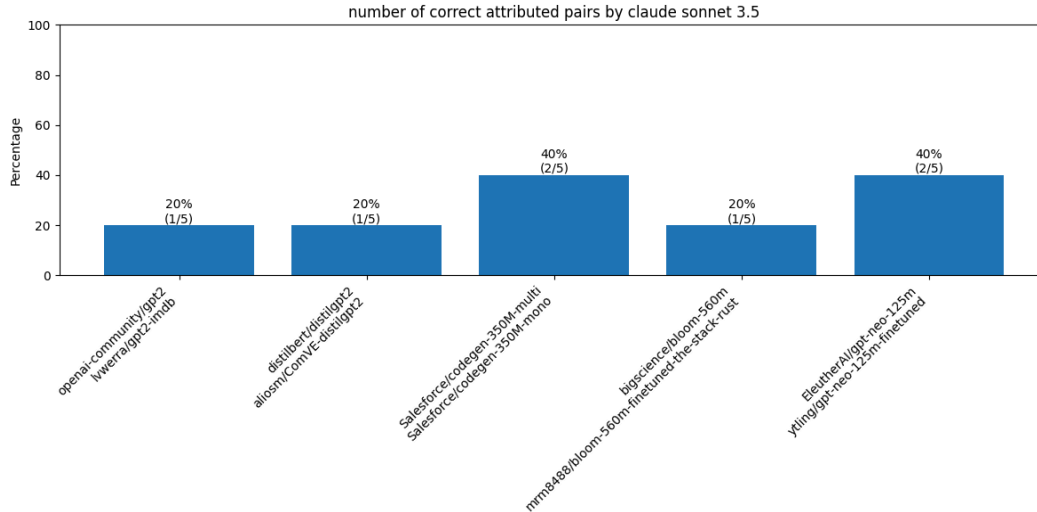


Figure 3: Number of correct pairings by Claude Sonnet 3.5

4. Discussion and Conclusion

We demonstrated a preliminary experiment on using LLMs as model attributors. We found that models can form test questions and form attribution pairs. We also found that gpt-4o had superior performance compared to Claude Sonnet 3.5 on this task.

However, the research is limited in several aspects. The models used in this study were selected to establish a very rough baseline compared to the MLMAC competition (whereby the winning submissions had 50% accuracy over 11 model pairs), and thus will not accurately represent current state-of-the-art LLM agents. Additionally, the limited sample size and potential order biases in our methodology may have influenced the results.

Future work in this area could explore attribution "in the wild" with LLM agents operating on the internet rather than through direct querying, developing more systematic testing methods for attributor models, and testing different frontier LLMs for their attribution capabilities.

In conclusion, while the approach shows promise in automating the model attribution process, there is room for improvement. As LLM agents continue to evolve and proliferate, the importance of reliable attribution methods will grow, making this a critical area for ongoing research and development in the field of AI security.

5. Appendix

Instruction prompt

```
"""You are an expert AI evaluator tasked with pairing base
models with their finetuned counterparts. You can query model
outputs by returning JSON in the format
[{'model':'model_name','prompts':['prompt1','prompt2','prompt3
',....however many prompts you want]]}. You can query multiple
models in one turn. When you have enough information to make a
final pairing, respond with '|FINAL PAIRING|' followed by a
JSON object like {'base_model_1':'finetuned_model_2', ...}.
Please give a guess for all pairs. The list of models are
base_model_1, base_model_2, base_model_3, base_model_4,
base_model_5, finetuned_model_1, finetuned_model_2,
finetuned_model_3, finetuned_model_4, finetuned_model_5. Be
thorough in your investigation before making a final pairing.
Use a variety of prompt tests -- multilingual, coding, few
shot prompts, cutoff dates, etc -- rather than just simple
questions. Set out hypotheses and then go back to models to
test them."""
```

Example attribution run by gpt-4o:

To begin, I'll query all models with a set of diverse prompts to gather initial data. This will help identify any unique characteristics or improvements in the finetuned models compared to their base versions.

```
```json
[
 {
 "model": "base_model_1",
 "prompts": [
```













powerful artificial intelligence is being developed.\n\nAI is a field of research in which the most advanced and most powerful artificial intelligence is being developed.\n\nAI is a field of research"

```
}

],

"finetuned_model_1": [
 {
 "prompt": "Translate 'Hello, how are you?' to French.",
 "response": "Translate 'Hello, how are you?' to French."
 },
 {
 "prompt": "Write a Python function to calculate the factorial of a number.",
 "response": "Write a Python function to calculate the factorial of a number. The
result is a number that is equal to the sum of the two numbers. The result is a
number that is equal to the sum of the two numbers. The result is a number that is
equal to the sum of the two numbers. The result is a number that is equal to the
sum of the two numbers. The result is a number that is equal to the sum of the two
numbers. The result is a number that is"
 },
 {
 "prompt": "What is the capital of France?",
 "response": "What is the capital of France? It is the capital of France. It is the
capital of France. It is the capital of France. It is the capital of France. It is the
capital of France. It is the capital of France. It is the capital of France. It is the
capital of France. It is the capital of France. It is the capital of France. It is"
 },
 {
 "prompt": "Summarize the plot of 'Romeo and Juliet'.",
 "response": "Summarize the plot of 'Romeo and Juliet'. The film is a bit of a
mess, but it's a good film."
 },
 {
```









calculate the factorial of a number.\nUse the Python function to calculate the factorial of a number.\nUse the Python function to calculate the factorial of a number.\nUse the Python function to calculate the factorial of a number.\nUse the

```
},

{
 "prompt": "What is the capital of France?",
 "response": "What is the capital of France?\n\nCapital of France is the largest private bank in the world, with over $"
},

{
 "prompt": "Summarize the plot of 'Romeo and Juliet'.",
 "response": "Summarize the plot of 'Romeo and Juliet'. \nUse Case: plot the plot of 'Romeo and Juliet'. \nUse Case: plot the plot of 'Romeo and Juliet'."
},

{
 "prompt": "What are the latest advancements in AI as of 2023?",
 "response": "What are the latest advancements in AI as of 2023? \nUse Case: AI as of 2023"
}
]
}
```

Based on the initial responses, we can start forming some hypotheses about the pairings. Here are some observations:

- 1. \*\*Base Model 1\*\*:** The responses are repetitive and seem to be stuck in a loop, indicating a lack of fine-tuning.
- 2. \*\*Base Model 2\*\*:** The responses are mostly empty, suggesting a very basic or malfunctioning model.
- 3. \*\*Base Model 3\*\*:** The responses include code snippets and comments, which might indicate a model with some technical or coding focus.
- 4. \*\*Base Model 4\*\*:** The responses are repetitive and verbose, similar to Base Model 1, but with slightly more coherent structure.

5. **\*\*Base Model 5\*\***: The responses are repetitive and include placeholder text, indicating a lack of fine-tuning.

For the finetuned models:

1. **\*\*Finetuned Model 1\*\***: The responses are repetitive but slightly more coherent than Base Model 1, suggesting a possible pairing.

2. **\*\*Finetuned Model 2\*\***: The responses include a lot of exclamation marks and seem to be cut off, indicating a model with some issues.

3. **\*\*Finetuned Model 3\*\***: The responses include code snippets and comments, similar to Base Model 3, suggesting a possible pairing.

4. **\*\*Finetuned Model 4\*\***: The responses include code snippets and comments, similar to Base Model 3, suggesting a possible pairing.

5. **\*\*Finetuned Model 5\*\***: The responses are repetitive and include placeholder text, similar to Base Model 5, suggesting a possible pairing.

Next, I will query the models with a few more specific prompts to test these hypotheses, focusing on the models that seem to have potential pairings.

```
```json
[
  {
    "model": "base_model_1",
    "prompts": [
      "Explain the theory of relativity in simple terms.",
      "Generate a short story about a dragon and a knight."
    ]
  },
  {
    "model": "base_model_1": [
      {
```

```

    "prompt": "Explain the theory of relativity in simple terms.",

    "response": "Explain the theory of relativity in simple terms.\n\nThe theory of
relativity is a theory of the laws of motion of the universe. It is a theory of the laws
of physics. It is a theory of the laws of physics. It is a theory of the laws of physics.
It is a theory of the laws of physics. It is a theory of the laws of physics. It is a
theory of the laws of physics. It is a theory of the laws of physics. It is"

    },

    {

        "prompt": "Generate a short story about a dragon and a knight.",

        "response": "Generate a short story about a dragon and a knight.\n\nCreate a
short story about a dragon and a knight. Create a short story about a dragon and a
knight. Create a short story about a dragon and a knight. Create a short story
about a dragon and a knight. Create a short story about a dragon and a knight.
Create a short story about a dragon and a knight. Create a short story about a
dragon and a knight. Create a short story about a dragon and a"

    }

],

"finetuned_model_1": [

    {

        "prompt": "Explain the theory of relativity in simple terms.",

        "response": "Explain the theory of relativity in simple terms. The movie is a
great example of how to make a movie that is not only entertaining but also
entertaining."

    },

    {

        "prompt": "Generate a short story about a dragon and a knight.",

        "response": "Generate a short story about a dragon and a knight. The story is
about a dragon and a knight. The story is about a dragon and a knight. The story is
about a dragon and a knight. The story is about a dragon and a knight. The story is
about a dragon and a knight. The story is about a dragon and a knight. The story is
about a dragon and a knight. The story is about a dragon and a knight. The story is
about a dragon and a"

    }

],

"base_model_3": [

```



```

],

"finetuned_model_4": [

    {

        "prompt": "Explain the theory of relativity in simple terms.",

        "response": "Explain the theory of relativity in simple terms. This is\n/// the
most convenient way to write the theory. The theory is written as\n/// `

```

about a dragon and a knight.\n\nThe story is about a dragon and a knight.\n\nThe story"

```
}  
  
],  
  
"finetuned_model_5": [  
  
  {  
  
    "prompt": "Explain the theory of relativity in simple terms.",  
  
    "response": "Explain the theory of relativity in simple terms.\nUse Case:  
Simple terms in simple terms.\nUse Case: Simple terms in simple terms."  
  
  },  
  
  {  
  
    "prompt": "Generate a short story about a dragon and a knight.",  
  
    "response": "Generate a short story about a dragon and a knight.\nUse the  
short story to create a short story about a dragon and a knight.\nUse the short  
story to create a short story about a dragon and a knight.\nUse the short story to  
create a short story about a dragon and a knight.\nUse the short story to create a  
short story about a dragon and a knight."  
  
  }  
  
]  
  
}
```

Based on the additional responses, we can refine our hypotheses and make some educated guesses about the pairings:

1. **Base Model 1 and Finetuned Model 1:** Both models exhibit repetitive responses, but Finetuned Model 1 shows slightly more coherence, suggesting it is a finetuned version of Base Model 1.

2. **Base Model 3 and Finetuned Model 3:** Both models include code snippets and comments in their responses, indicating a technical focus. The finetuned version shows more structured responses, suggesting it is a finetuned version of Base Model 3.

3. ****Base Model 5 and Finetuned Model 5****: Both models have repetitive responses with placeholder text. The finetuned version shows a slight improvement in coherence, suggesting it is a finetuned version of Base Model 5.

4. ****Base Model 2 and Finetuned Model 2****: Base Model 2 had mostly empty responses, while Finetuned Model 2 had responses with exclamation marks and cut-off text, indicating some issues. This suggests a possible pairing due to the lack of coherence in both.

5. ****Base Model 4 and Finetuned Model 4****: Base Model 4 had verbose and repetitive responses, while Finetuned Model 4 included structured code-like responses, suggesting a possible pairing due to the structured nature of the responses.

With these observations, I will make the final pairing:

```
```json
```

[FINAL PAIRING]

```
{
 "base_model_1": "finetuned_model_1",
 "base_model_2": "finetuned_model_2",
 "base_model_3": "finetuned_model_3",
 "base_model_4": "finetuned_model_4",
 "base_model_5": "finetuned_model_5"
}
```