Do Machines and Humans Focus on Similar Code?

Exploring Explainability of Large Language Models *in Code Summarization*

Presented by Jiliang Li (jiliang.li@vanderbilt.edu)

Jiliang Li
Vanderbilt University

Yifan Zhang
Vanderbilt University

Zachary Karas
Vanderbilt University

Collin McMillan
University of Notre Dame

Kevin Leach
Vanderbilt University

Yu Huang
Vanderbilt University

VANDERBILT School *of* Engineering

UNIVERSITY OF NOTRE DAME College of Engineering

Large Language Models for Code

One indicative tasks for LLMs to comprehend code is

neural code summarization

– generating natural language summaries describing a code snippet.

You

Please write a brief summary describing what the method below is doing

```
public static void helloWorld() {
   System.out.println("Hello, World!");
}
```

ChatGPT

The method `helloWorld` is a simple Java method designed to display the message "Hello, World!" on the console.

https://arxiv.org/abs/2308.12950

Large Language Models for Code

However, we lack a formulaic or intuitive understanding of what and how models learn from code.

Explainability

- ❑ Improve model architecture
- ❑ Reducing bias
- ❑ Preventing undesired behaviors
- ❑ …

You

Hey Chat, what and how do you learn from the following code snippet?

```
def hello_world():
  print('hello world')
```

ChatGPT

?

https://arxiv.org/abs/2308.12950

Overview

Is there a general correlation between human and machine focus patterns for code summarization?

Do the code summaries increase in quality when machine focus becomes more aligned with that of humans?

Overview

Hypothesis → **Approach** → **Negative Result**

Sufficiently large models may learn to focus on parts of code similarly to humans.
If validated, we may provide partial AI explainability through the lens of human behaviors.

Overview

Hypothesis → **Approach** → **Negative Result**

Hypothesis

Sufficiently large models may learn to focus on parts of code similarly to humans.
If validated, we may provide partial AI explainability through the lens of human behaviors.

Approach

Approximate programmers' visual focus using an **eye-tracker.**

Approximate language model's focus using **SHapley Additive exPlanations**.

Negative Result

Overview

Hypothesis

Sufficiently large models may learn to focus on parts of code similarly to humans.
If validated, we may provide partial AI explainability through the lens of human behaviors.

Approach

Approximate programmers' visual focus using an **eye-tracker.**

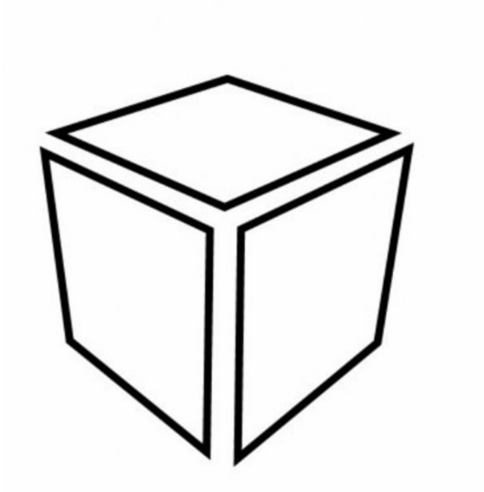
Approximate language model's focus using **SHapley Additive exPlanations**.

Negative Result

Using such approaches, language models' focus exhibits *NO* statistically significant correlation with human focus in general.

Interpreting Language Models

White Box



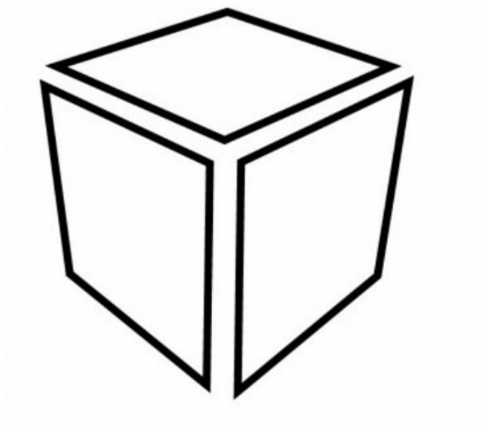
VS

Black Box

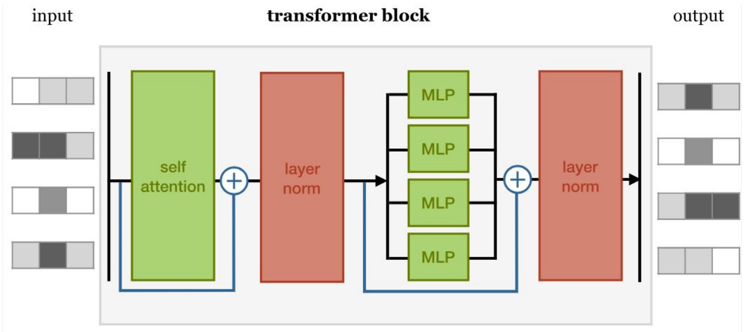
https://medium.com/@tam.tamanna18/comparing-black-box-vs-white-box-modeling-bd01575b7670

Interpreting Language Models

White Box



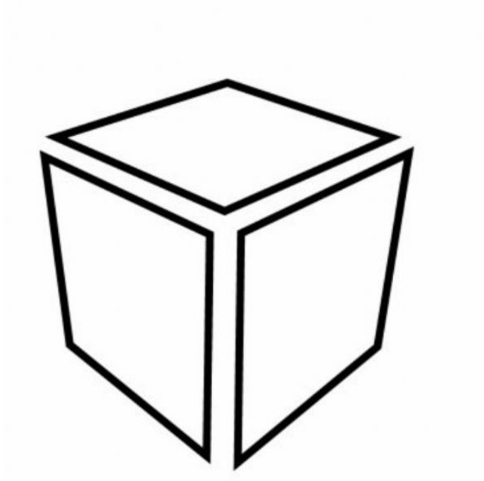
Requires access to internal layers



https://peterbloem.nl/blog/transformers

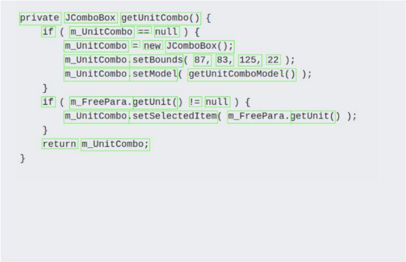
Interpreting Language Models

White Box

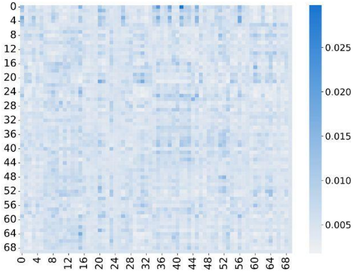


Requires access to internal layers

Code Snippet



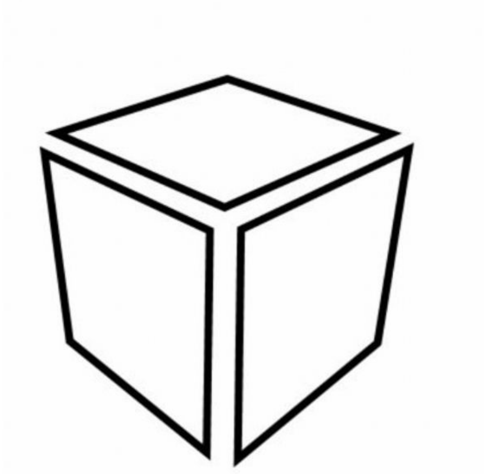
Self-Attention Matrix



https://arxiv.org/abs/2402.14096

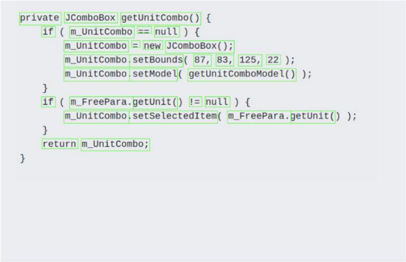
Interpreting Language Models

White Box

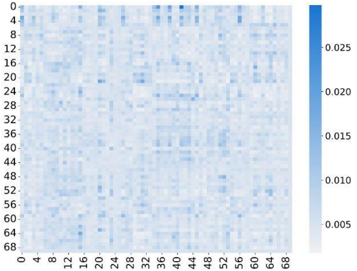


Precludes proprietary models

Code Snippet



Self-Attention Matrix



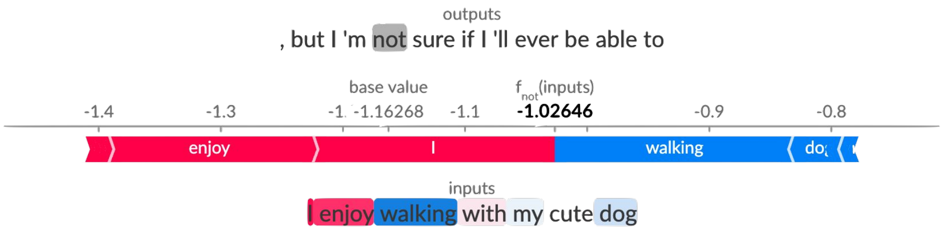
https://arxiv.org/abs/2402.14096

Interpreting Language Models

SHAP

SHapley Additive exPlanations

❑ Model-Architecture Agnostic



Black Box

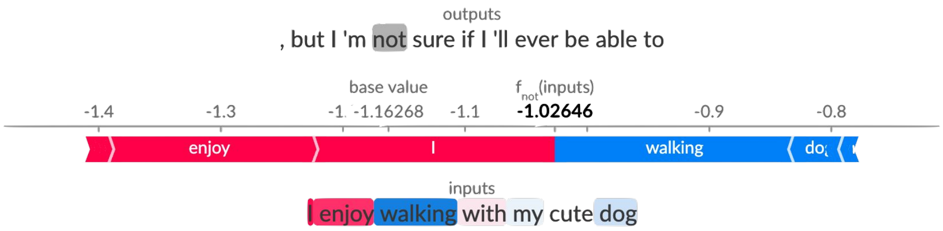
https://shap.readthedocs.io/en/latest/example_notebooks/text_examples/text_generation/Open%20Ended%20GPT2%20Text%20Generation%20Explanations.html

Interpreting Language Models

SHAP

SHapley Additive exPlanations

❏ Model-Architecture Agnostic



Q: How much does each input feature contribute to the output?

A: Apply game-theoretic principles to assess how each input features' presence / absence (simulated by token masking) alters the model's prediction from the expected result.

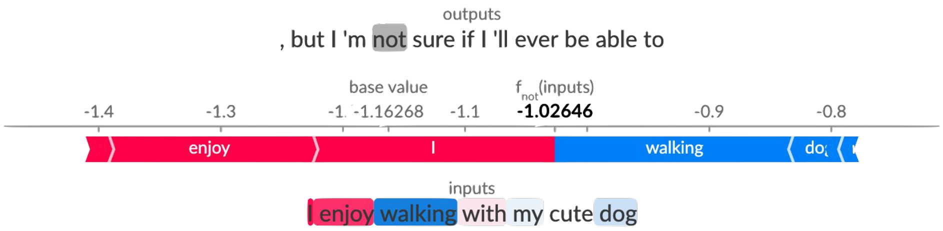
https://shap.readthedocs.io/en/latest/example_notebooks/text_examples/text_generation/Open%20Ended%20GPT2%20Text%20Generation%20Explanations.html

Interpreting Language Models

SHAP

SHapley Additive exPlanations

❏ Model-Architecture Agnostic



Q: How much does each input feature contribute to the output?

A: Measure *feature attribution*, i.e. which parts of code are most important for the model to generate its desired output.

https://shap.readthedocs.io/en/latest/example_notebooks/text_examples/text_generation/Open%20Ended%20GPT2%20Text%20Generation%20Explanations.html

Neural Code Summarization

6
Language
Models

Neural Code Summarization

GPT-4

GPT-few -shot

GPT-3.5

StarCoder 15.5B

CodeLlama-Instruct 7B

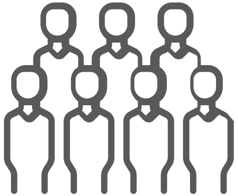
NCS

A GPT-3.5 model, but instructed via few-shot prompting to generate summaries more similar to human summaries.

"A Transformer-based Approach for Source Code Summarization," Ahmad et al. (2020), first transformer model for code summarization.

6 Language Models

Human vs Neural Code Summarization



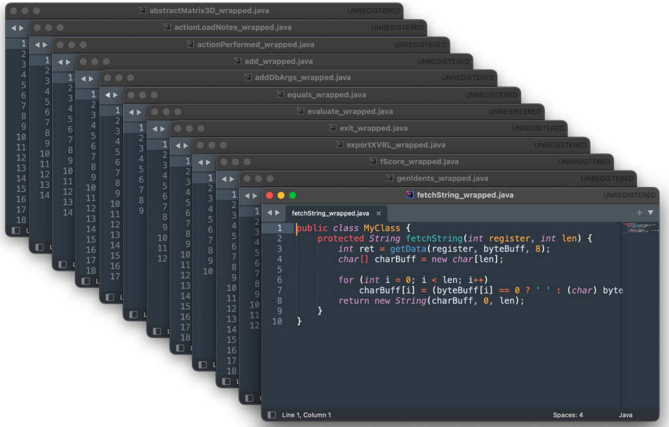
27
Programmers

6
Language
Models

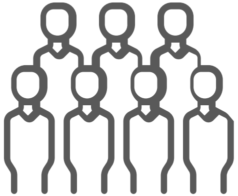
Human vs Neural Code Summarization

162
Java Methods

from the
FunCom dataset



summarize



27
Programmers

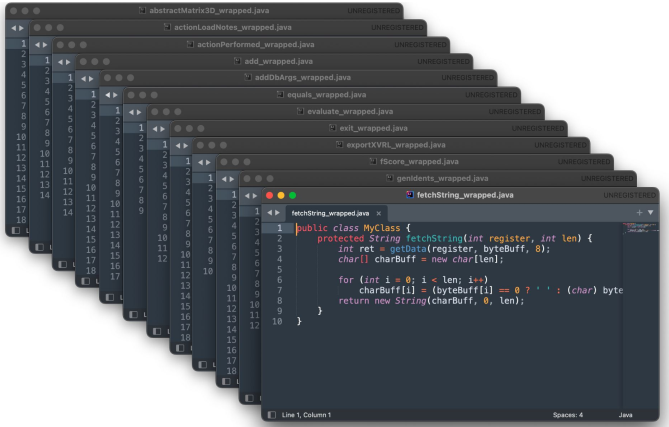
6
Language
Models

Human vs Neural Code Summarization

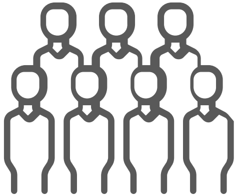
Java

162
Java Methods

from the
FunCom dataset



summarize



27
Programmers

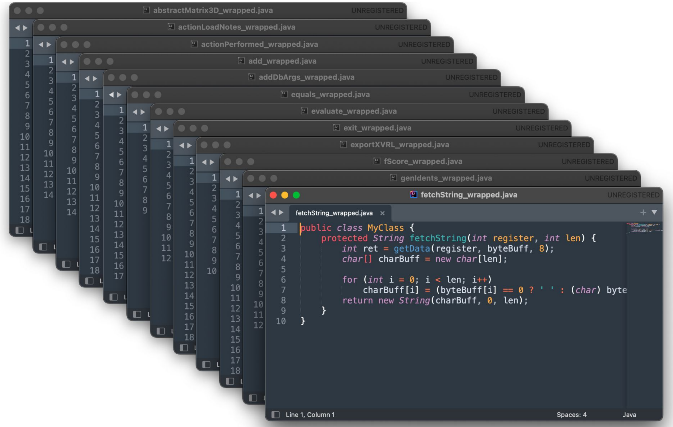
compare
focus

6
Language
Models

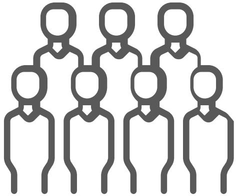
Human Code Summarization

162
Java Methods

from the
FunCom dataset



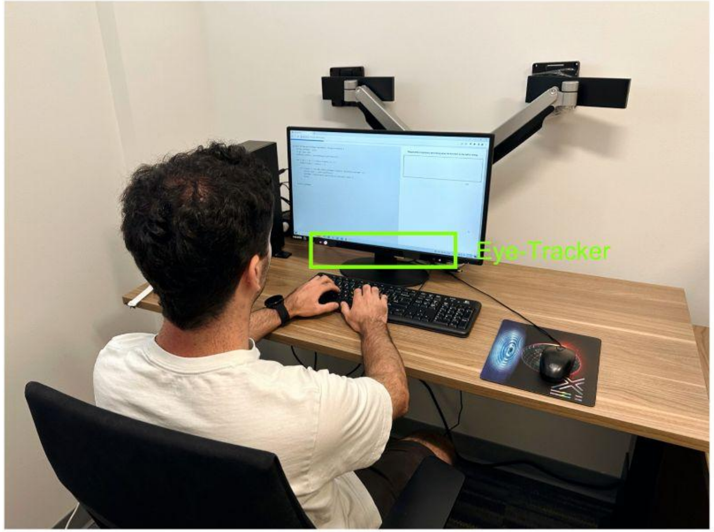
summarize

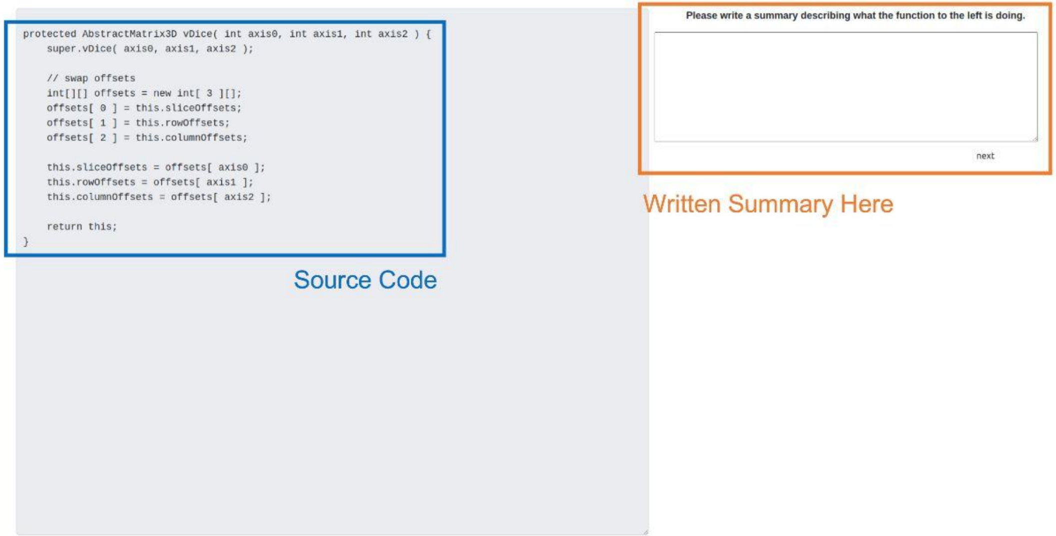


27
Programmers

- Each summarized 24-25 methods
- 671 trials of eye-tracking data
- 5 low-quality data points removed

Measuring Human Visual Focus





Tobii Pro Fusion Eye-Tracker

Java Method

Summary Writing

Experimental Room

Example Task

Comparing Human and Model Foci

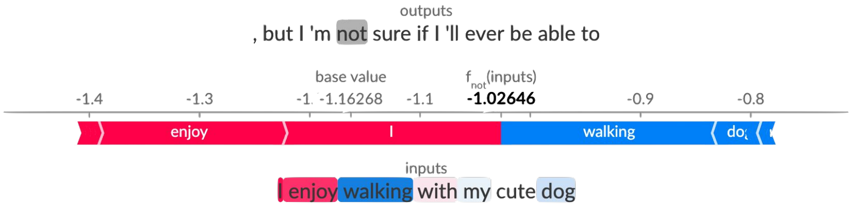
Human Focus

- ❏ Fixation Duration
- ❏ Fixation Count

A fixation is a spatially stable eye-movement lasting 100-300ms

Machine Focus

- ❏ SHAP



- ❏ Each Abstract Syntax Tree (AST) token in each Java method is the basic unit of focus calculation.
- ❏ Focus scores normalized across each method.

Comparing Human and Model Foci

```
[ public, void, helloWorld, System, out, println, \”Hello, World!\” ]
```

- ❏ Each Abstract Syntax Tree (AST) token in each Java method is the basic unit of focus calculation.
- ❏ Focus scores normalized across each method.

Comparing Human and Model Foci

Human Focus
Fixation Count

```
[0.11, 0.14, 0.18, 0.11, 0.11, 0.18, 0.18]
```

```
[ public, void, helloWorld, System, out, println, \"Hello, World!\" ]
```

- ❏ Each Abstract Syntax Tree (AST) token in each Java method is the basic unit of focus calculation.
- ❏ Focus scores normalized across each method.

Comparing Human and Model Foci

Human Focus
Fixation Count

GPT-4 Focus
SHAP

```
[0.11, 0.14, 0.18, 0.11, 0.11, 0.18, 0.18]
[0.11, 0.15, 0.19, 0.07, 0.15, 0.19, 0.15]
```

```
[ public, void, helloWorld, System, out, println, \"Hello, World!\" ]
```

- ❏ Each Abstract Syntax Tree (AST) token in each Java method is the basic unit of focus calculation.
- ❏ Focus scores normalized across each method.

Comparing Human and Model Foci

Spearman's rank correlation coefficient ρ

```
ρ = 0.808, p = 0.028
```

Human Focus
Fixation Count

GPT-4 Focus
SHAP

```
[0.11, 0.14, 0.18, 0.11, 0.11, 0.18, 0.18]
```

```
[0.11, 0.15, 0.19, 0.07, 0.15, 0.19, 0.15]
```

```
[ public, void, helloWorld, System, out, println, \"Hello, World!\" ]
```

- ❏ Each Abstract Syntax Tree (AST) token in each Java method is the basic unit of focus calculation.
- ❏ Focus scores normalized across each method.

RQ1

Is there a general correlation between human and machine focus patterns for code summarization?

- ❏ For each pair of focus sources amongst

  {Fixation Count, Fixation Duration, GPT-4, GPT-few-shot, GPT-3.5, StarCoder, CodeLlama, NCS}

  We report the means and standard deviations of Spearman's ρ for all Java methods showing significant correlation ($p \leq 0.05$).

Human vs Machine Foci across Java Methods

	Duration	Count	GPT4	GPT-few-shot	GPT3.5	StarCoder	Code Llama	NCL
Duration	1.00±0.00	0.88±0.06	-0.11±0.41	-0.13±0.42	-0.09±0.52	-0.18±0.48	-0.18±0.42	-0.24±0.40
Count	-	1.00±0.00	0.01±0.45	-0.24±0.33	-0.10±0.48	-0.31±0.29	-0.13±0.43	-0.33±0.33
GPT4	-	-	1.00±0.00	0.68±0.12	0.76±0.12	0.67±0.14	0.67±0.14	0.55±0.13
GPT-few-shot	-	-	-	1.00±0.00	0.72±0.12	0.62±0.15	0.64±0.15	0.55±0.13
GPT3.5	-	-	-	-	1.00±0.00	0.65±0.16	0.67±0.15	0.58±0.13
StarCoder	-	-	-	-	-	1.00±0.00	0.66±0.15	0.59±0.11
Code Llama	-	-	-	-	-	-	1.00±0.00	0.56±0.14
NCL	-	-	-	-	-	-	-	1.00±0.00

The means and standard deviations of Spearman's correlation (ρ) between human and model foci, collected from all Java methods showing significant correlation ($p \le 0.05$).

Human vs Machine Foci across Java Methods

Human Focus

Machine Focus

	Duration	Count	GPT4	GPT-few-shot	GPT3.5	StarCoder	Code Llama	NCL
Duration	1.00±0.00	0.88±0.06	-0.11±0.41	-0.13±0.42	-0.09±0.52	-0.18±0.48	-0.18±0.42	-0.24±0.40
Count	-	1.00±0.00	0.01±0.45	-0.24±0.33	-0.10±0.48	-0.31±0.29	-0.13±0.43	-0.33±0.33
GPT4	-	-	1.00±0.00	0.68±0.12	0.76±0.12	0.67±0.14	0.67±0.14	0.55±0.13
GPT-few-shot	-	-	-	1.00±0.00	0.72±0.12	0.62±0.15	0.64±0.15	0.55±0.13
GPT3.5	-	-	-	-	1.00±0.00	0.65±0.16	0.67±0.15	0.58±0.13
StarCoder	-	-	-	-	-	1.00±0.00	0.66±0.15	0.59±0.11
Code Llama	-	-	-	-	-	-	1.00±0.00	0.56±0.14
NCL	-	-	-	-	-	-	-	1.00±0.00

The means and standard deviations of Spearman’s correlation (ρ) between human and model foci, collected from all Java methods showing significant correlation ($p \leq 0.05$).

Human vs Machine Foci across Java Methods

Correlation coefficients have small means and large standard deviations.

	Duration	Count	GPT4	GPT-few-shot	GPT3.5	StarCoder	Code Llama	NCL
Duration	1.00±0.00	0.88±0.06	-0.11±0.41	-0.13±0.42	-0.09±0.52	-0.18±0.48	-0.18±0.42	-0.24±0.40
Count	-	1.00±0.00	0.01±0.45	-0.24±0.33	-0.10±0.48	-0.31±0.29	-0.13±0.43	-0.33±0.33
GPT4	-	-	1.00±0.00	0.68±0.12	0.76±0.12	0.67±0.14	0.67±0.14	0.55±0.13
GPT-few-shot	-	-	-	1.00±0.00	0.72±0.12	0.62±0.15	0.64±0.15	0.55±0.13
GPT3.5	-	-	-	-	1.00±0.00	0.65±0.16	0.67±0.15	0.58±0.13
StarCoder	-	-	-	-	-	1.00±0.00	0.66±0.15	0.59±0.11
Code Llama	-	-	-	-	-	-	1.00±0.00	0.56±0.14
NCL	-	-	-	-	-	-	-	1.00±0.00

The means and standard deviations of Spearman's correlation (ρ) between human and model foci, collected from all Java methods showing significant correlation ($p \leq 0.05$).

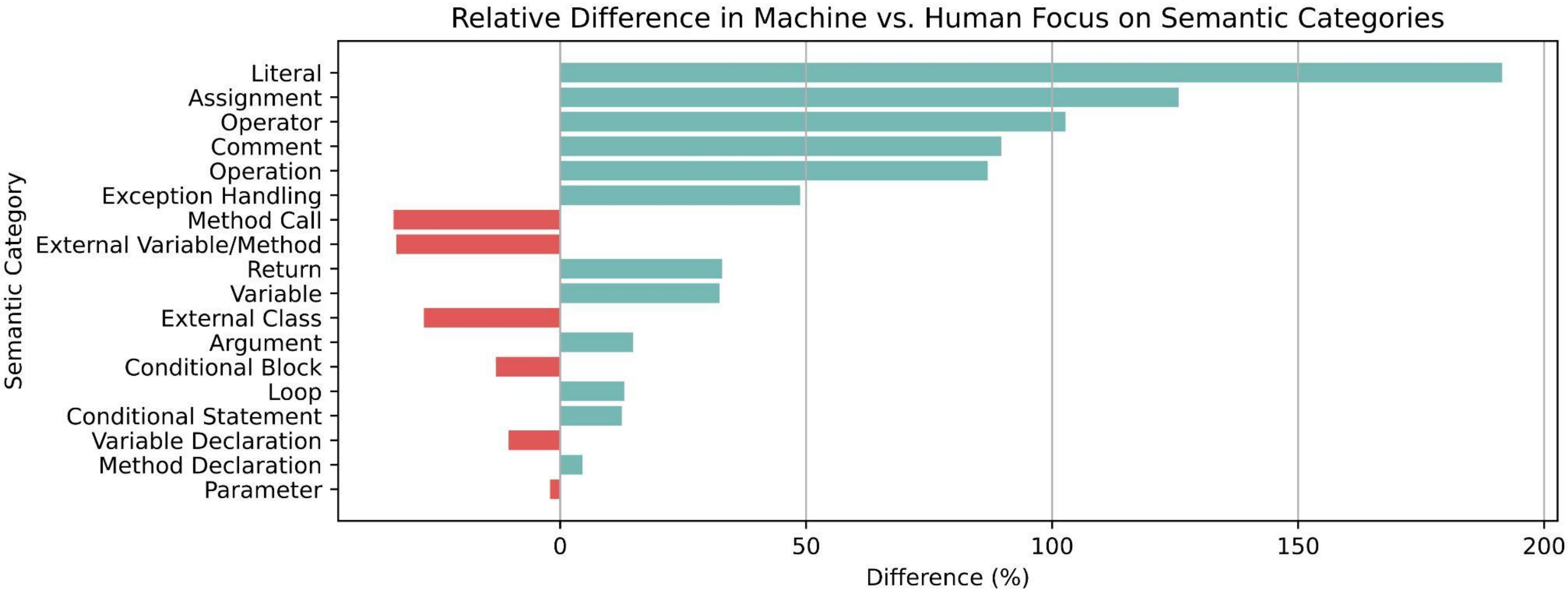
Human vs Machine Foci across Java Methods

Correlation coefficients have small means and large standard deviations.

★ Correlation between human and machine foci varies significantly depending on the specific Java method.

★ No correlation between human and machine foci is widespread across all methods.

Where do Human and Machine Focus on?



RQ2

Do the code summaries increase in quality when machine focus becomes more aligned with that of humans?

- ❏ A human expert provides quality ratings for summaries generated by each language model.
- ❏ Compute correlations between a model's *summary quality* and how well its focus *aligns* with humans'.

Human-machine Focus Alignment vs Quality

- Four metrics – Accuracy, Completeness, Conciseness, Readability – used to assess machine-generated code summary quality, each rated on a scale from 1-4.

How well does human-machine focus alignment correlate with summary quality?

Human-machine Focus Alignment vs Quality

- ❑ Four metrics – Accuracy, Completeness, Conciseness, Readability – used to assess machine-generated code summary quality, each rated on a scale from 1-4.

How well does human-machine focus alignment correlate with summary quality?

	Accuracy	Completeness	Conciseness	Readability
Spearman's ρ	-0.1279	0.1309	0.0194	-0.0717
p-value	0.3862	0.3753	0.8960	0.6280

Correlation coefficients are small and p-values are large.

Human-machine Focus Alignment vs Quality

★ Regardless of which metric is used to assess code summary quality, there is a lack of statistically significant correlation between

the quality of a model-generated summary

and

how well the model's focus aligns with humans' on that Java method.

Correlation coefficients are small and p-values are large.

Human-machine Focus Alignment vs Quality

- ★ Regardless of which metric is used to assess code summary quality, there is a lack of statistically significant correlation between

  the quality of a model-generated summary

  and

  how well the model's focus aligns with humans' on that Java method.

- ★ Aspects other than feature attribution are possibly more indicative of and critical to language model's performance in code summarization.

Possible Interpretations

1 Possible Difference

It is possible that language models and humans reason about code differently when summarizing source code.

Possible Interpretations

1 Possible Difference

It is possible that language models and humans reason about code differently when summarizing source code.

2 Call for Alternatives

Alternative methods may be needed to assess feature influence in black-box language models for code summarization, aiming for better alignment with human attention.

Possible Interpretations

1 Possible Difference

It is possible that language models and humans reason about code differently when summarizing source code.

2 Call for Alternatives

Alternative methods may be needed to assess feature influence in black-box language models for code summarization, aiming for better alignment with human attention.

3 Call for Access

Access to the internal workings of proprietary models might become critical if white-box models offer more human-aligned insights into explainable language models for code [Paltenghi et al. (2022)].

Remarks

We contain our conclusion to the SHAP measure of feature attribution and the human attention as measured in an eye-tracking experiment.

We contribute with our finding that SHAP did not correlate with human eye attention in the measures or models we studied.

Summary

- **Experimental Design**: 27 programmers and 6 LLMs tasked to summarize 162 Java methods; eye-tracking fixation to approximate attention of programmers; SHAP feature attribution as a proxy to measure LLMs' focus on code.
- **RQ1 Result**: Correlation between human and machine foci varies significantly depending on which specific Java method the programmers/LLMs are tasked to summarize.
- **RQ2 Result**: There is a lack of statistically significant correlation between the quality of a model-generated summary and how well the model's focus aligns with humans'.
- **Conclusion**: Using SHAP to approximate feature attribution does not provide explainability of language models through establishing correlations between machine and human foci.

Large Language Models for Code

Large Language Models for code have demonstrated proficiency at code comprehension.

Model	Size	HumanEval			MBPP		
		pass@1	pass@10	pass@100	pass@1	pass@10	pass@100
code-cushman-001	12B	33.5%	-	-	45.9%	-	-
GPT-3.5 (ChatGPT)	-	48.1%	-	-	52.2%	-	-
GPT-4	-	67.0%	-	-	-	-	-
PaLM	540B	26.2%	-	-	36.8%	-	-
PaLM-Coder	540B	35.9%	-	88.4%	47.0%	-	-
PaLM 2-S	-	37.6%	-	88.4%	50.0%	-	-
StarCoder Base	15.5B	30.4%	-	-	49.0%	-	-
StarCoder Python	15.5B	33.6%	-	-	52.7%	-	-
StarCoder Prompted	15.5B	40.8%	-	-	49.5%	-	-
Llama 2	7B	12.2%	25.2%	44.4%	20.8%	41.8%	65.5%
	13B	20.1%	34.8%	61.2%	27.6%	48.1%	69.5%
	34B	22.6%	47.0%	79.5%	33.8%	56.9%	77.6%
	70B	30.5%	59.4%	87.0%	45.4%	66.2%	83.1%
Code Llama	7B	33.5%	59.6%	85.9%	41.4%	66.7%	82.5%
	13B	36.0%	69.4%	89.8%	47.0%	71.7%	87.1%
	34B	48.8%	76.8%	93.0%	55.0%	76.2%	86.6%
	70B	53.0%	84.6%	96.2%	62.4%	81.1%	91.9%
Code Llama - Instruct	7B	34.8%	64.3%	88.1%	44.4%	65.4%	76.8%
	13B	42.7%	71.6%	91.6%	49.4%	71.2%	84.1%
	34B	41.5%	77.2%	93.5%	57.0%	74.6%	85.4%
	70B	67.8%	90.3%	97.3%	62.2%	79.6%	89.2%
Unnatural Code Llama	34B	62.2%	85.2%	95.4%	61.2%	76.6%	86.7%
Code Llama - Python	7B	38.4%	70.3%	90.6%	47.6%	70.3%	84.8%
	13B	43.3%	77.4%	94.1%	49.0%	74.0%	87.6%
	34B	53.7%	82.8%	94.7%	56.2%	76.4%	88.2%
	70B	57.3%	89.3%	98.4%	65.6%	81.5%	91.9%

https://arxiv.org/abs/2308.12950

Large Language Models for Code

However, we lack a formulaic or intuitive understanding of what and how models learn from code.

You

Hey Chat, what and how do you learn from the following code snippet?

```
def hello_world():
    print('hello world')
```

ChatGPT

?

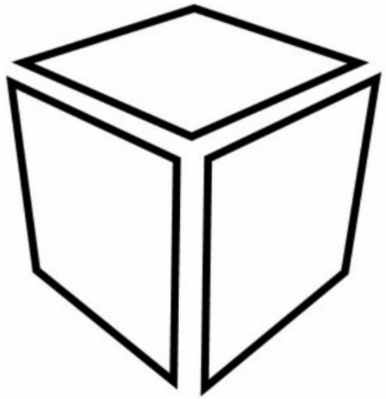
Explainability

- ❏ Improve model architecture
- ❏ Reducing bias
- ❏ Preventing undesired behaviors
- ❏ ...

https://arxiv.org/abs/2308.12950

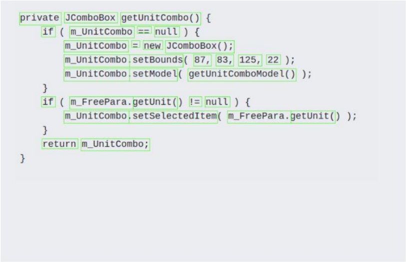
Interpreting Language Models

White Box

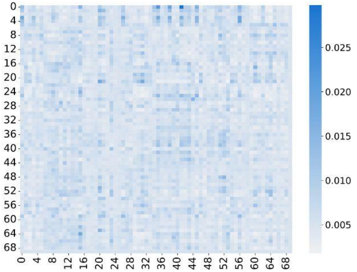


But attention weights are difficult to aggregate

Code Snippet



Self-Attention Matrix



https://arxiv.org/abs/2402.14096

RQ1

Correlation coefficients have small means and large standard deviations.

	Duration	Count	GPT4	GPT-few-shot	GPT3.5	StarCoder	Code Llama	NCL
Duration	1.00±0.00	0.88±0.06	-0.11±0.41	-0.13±0.42	-0.09±0.52	-0.18±0.48	-0.18±0.42	-0.24±0.40
Count	-	1.00±0.00	0.01±0.45	-0.24±0.33	-0.10±0.48	-0.31±0.29	-0.13±0.43	-0.33±0.33
GPT4	-	-	1.00±0.00	0.68±0.12	0.76±0.12	0.67±0.14	0.67±0.14	0.55±0.13
GPT-few-shot	-	-	-	1.00±0.00	0.72±0.12	0.62±0.15	0.64±0.15	0.55±0.13
GPT3.5	-	-	-	-	1.00±0.00	0.65±0.16	0.67±0.15	0.58±0.13
StarCoder	-	-	-	-	-	1.00±0.00	0.66±0.15	0.59±0.11
Code Llama	-	-	-	-	-	-	1.00±0.00	0.56±0.14
NCL	-	-	-	-	-	-	-	1.00±0.00

These are values are only calculated from Java methods where Spearman's ρ is statistically significant ($p \leq 0.05$). Such Java methods only constitute 22% of all methods.

RQ1

Correlation coefficients have small means and large standard deviations.

★ Correlation between human and machine foci varies significantly depending on the specific Java method.

★ No correlation between human and machine foci is widespread across all methods.

These are values are only calculated from Java methods where Spearman's ρ is statistically significant ($p \leq 0.05$). Such Java methods only constitute 22% of all methods.

RQ1

Some other findings:

- ❑ GPT-few-shot generates summaries much more similar to humans', their focus is not more correlated with humans'.

- ❑ Feature attribution (SHAP values) in all language models is moderately or strongly correlated with each other. This intuitively makes sense since all six models studied are based on the Transformer architecture.

Neural Code Summarization

162
Java Methods

from the
FunCom dataset

summarize

6
Language
Models