CROSS-LINGUAL PERFORMANCE OF **CODEGPT** ON THE **CODE COMPLETION** TASK

Nadine Kuo
h.n.kuo@student.tudelft.nl

Supervisors: **Maliheh Izadi, Jonathan Katzy**
Professor: **Arie van Deursen**

1. Introduction

- **Code intelligence** tools such as *GitHub Copilot* have significantly enabled developers to enhance productivity and efficiency [1].
- These tools are based on **Large Language Models (LLMs)** that have been trained on source code in order to perform programming-related tasks including **code completion.**

- However most of these models are **trained** on merely widely-used programming languages, which may limit the performance on **low-resource languages.**
- This also means there is **limited research on performance** of code models on **low-resource languages**, inspiring us to....

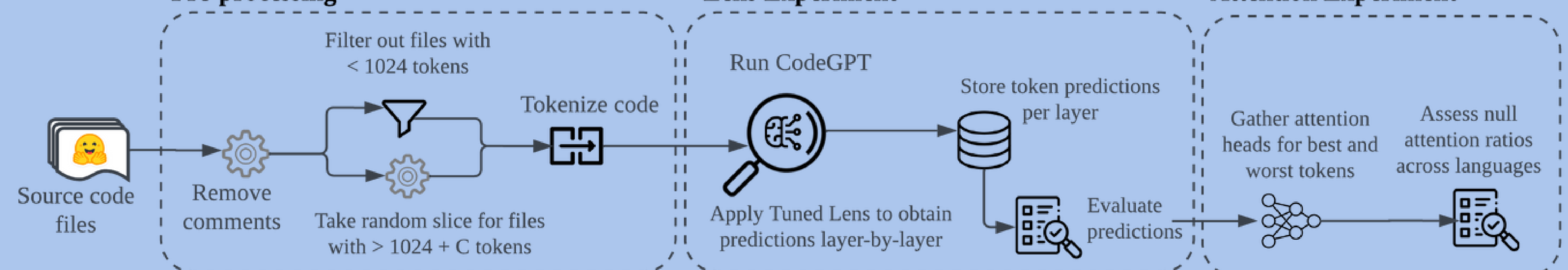
investigate how the GPT-2-based Transformer **CodeGPT** performs on **the token-level code completion task** across **high-** and **low-resource languages.**

*Note that our multilingual CodeGPT model was fine-tuned on: Java, JS, Python, PHP, Go and Ruby. These will be indicated by * .*

2. Research Questions & Methodology

RQ1: When does CodeGPT generate *incorrect* tokens with *high confidence* ("worst" predictions)?

RQ2: Cross-lingual patterns in CodeGPT's attention mechanism?



100K files per language adapted from The Stack [2]: permissively licensed GitHub repositories

Figure 1: Toy example of intermediate token predictions (lens output) for Java (left) and Kotlin (right)

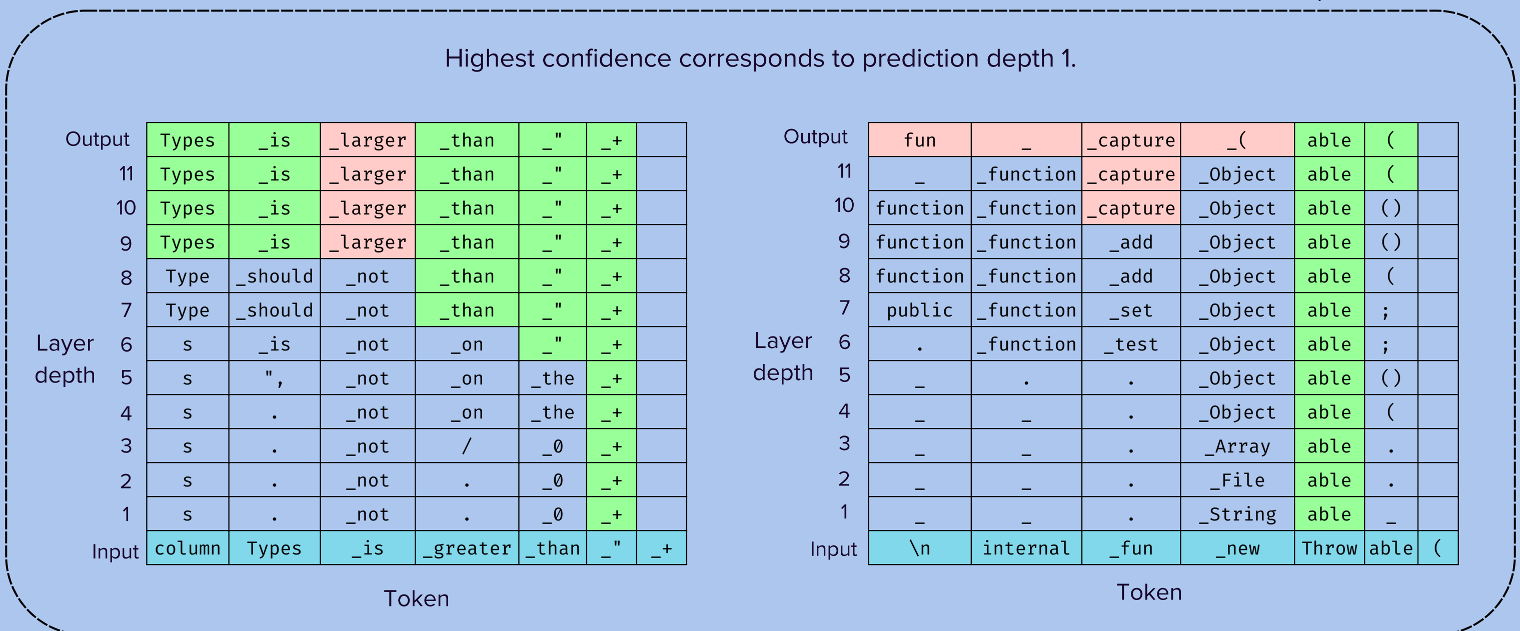


Figure 2: Visualization of null attention using Bertviz [3]

$$\text{Null Attention Ratio} = \frac{1}{N}\sum_{i=0}^{N} \alpha_{i,0}(x) \geq 0.9$$

CodeGPT:
- *12 layers*
- *12 attention heads per layer*

3. Results

Best and Worst Predictions

Figure 3: Confidence of False Predictions

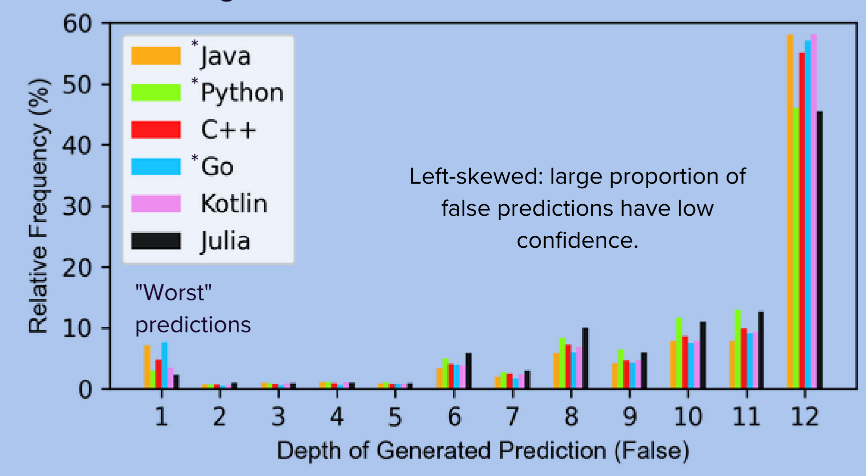
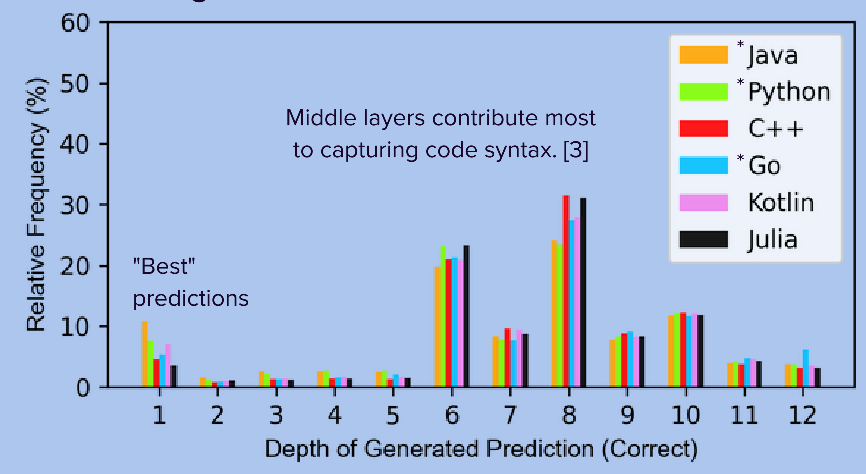


Figure 4: Confidence of Correct Predictions



Best
- **Kotlin, Go**: mostly user-defined elements
  - Reliance on inherited natural language understanding through GPT-2
  - Endings of conjunctions are trivial due to uni-directional architecture
- **Java, Python, C++ and Go**: also contain common code structures and language-specific elements

Worst
- **Kotlin, Go**: beyond punctuation , also common code structures and language-specific elements
- **Java, Python, C++ and Go**: structures for which left-context is not sufficient, including punctuation

*Java		*Python		C++		*Go		Kotlin		Julia	
Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst
get	}	self	def	;	}	x	}	get	}	1	end
;	public	s	1	x	if	:	or	s	return	0	1
<	return	get	class	get	{	n	?	qual	if	2	2
als	if	0	:	n	->	for	!	s	get	t	@
\\	@	1	=	0	return	n	<	ator	"	[	[
Exception	VK	if	if	1	#	!!	case	n	get	ices	:
String	private	n	@	s	::	son	<	els	"	[	[
or	assert	'	[	int	case	s	is	point	get	n	{
of	this	format	2	or	void	n	java	1	function	]	else
adata	{	args	0	r	log	ert	to	els	var	estamp	]

Table 1: Top-10 tokens predicted tokens with highest confidence

Null Attention Patterns

- ***Distributions*** **of confidence of correct predictions (Figure 4)** and ***magnitude of*** **null attention (Figure 5)** align with each other.
- **Performance (Table 2)** correlates positively with ***magnitude of*** **null attention ratios (Table 3).**
- **Model confidence** correlates positively with ***variance of*** **null attention** across model layers **(Table 3).**
- *Magnitude and variance* of null attention are **similar** between:
  - **Best and worst** predictions
  - Token **categories**

	EM (%)	MRR (%)
*Java	69.2	16.9
*Python	68.2	14.7
C++	64.5	11.6
*Go	67.9	12.4
Kotlin	58.3	11.8
Julia	65.0	11.2

Table 2: Cross-lingual Performance

	Mean ± SD	Mean ± SD in confident setting
*Java	9.3 ± 3.8	10.3 ± 8.7
*Python	8.9 ± 3.4	8.2 ± 7.9
C++	9.2 ± 3.5	9.3 ± 9.3
*Go	9.1 ± 3.1	8.6 ± 8.1
Kotlin	8.2 ± 2.7	7.8 ± 7.6
Julia	7.7 ± 2.7	8.6 ± 9.7

Table 3: Null Attention Ratio Statistics over Layers 6-12

Figure 5: Overall Null Attention Ratios by Layers

4. Conclusion

	Familiar languages	Unfamiliar languages
Best	Common code structures Language-specific elements	User-defined elements
Worst	Structures for which unidirectional architecture is insufficient	Common code structures Language-specific elements

Highest confidence and accuracy

Does not recognize language-specific code syntax

- Positive correlation between model **accuracy** and ***magnitude of*** **null attention**
- Positive correlation between model **confidence** and ***variance of*** **null attention across layers**

Future research

- **Null attention is not sufficient** to explain cross-lingual differences between **best and worst** tokens nor token **categories**
- Investigate differences in **attention patterns** (beyond null attention) across high- and low-resource languages
  - Idea: **clustering of attention heads** to group common patterns or detect unique patterns

[1] Radford, A. et al. Language models are unsupervised multitask learners. OpenAI blog, 9 (2019).
[2] Kocetkov, D. et al. (2022, November 20). The Stack: 3 TB of permissively licensed source code.
[3] Vig, J. (2019, July). A Multiscale Visualization of Attention in the Transformer Model.

TUDelft