HumanEval-XL: A Multilingual Code Generation Benchmark for Cross-lingual Natural Language Generalization

UNIVERSITY OF COPENHAGEN   Bai百度

Qiwei Peng*, Yekun Chai*, Xuhong Li

University of Copenhagen   Baidu Inc.

1 Code Generation

Given Input:

```python
def binomial_coeff(n,k):
    """ Write a Python function to
find binomial coefficient.
    >>> binomial_coeff(5,2)
    10
    """
```

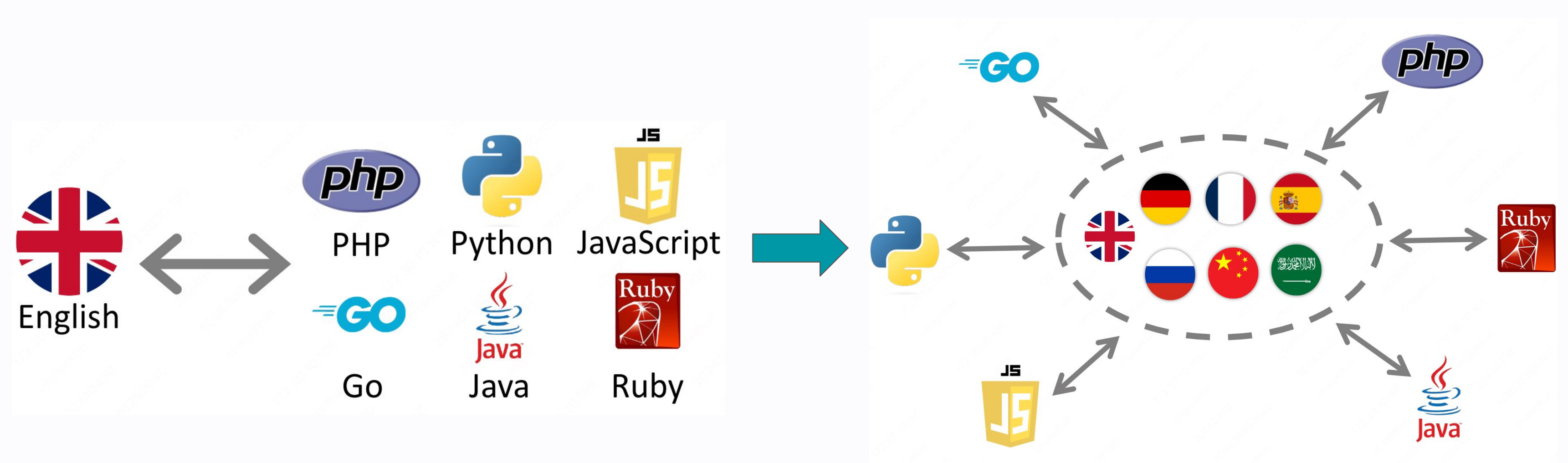
Model →

Expected Output:

```python
    if k > n :
       return 0
    if k==0 or k ==n :
        return 1
    return binomial_coeff(n-1,k-1)
+ binomial_coeff(n-1,k)
```

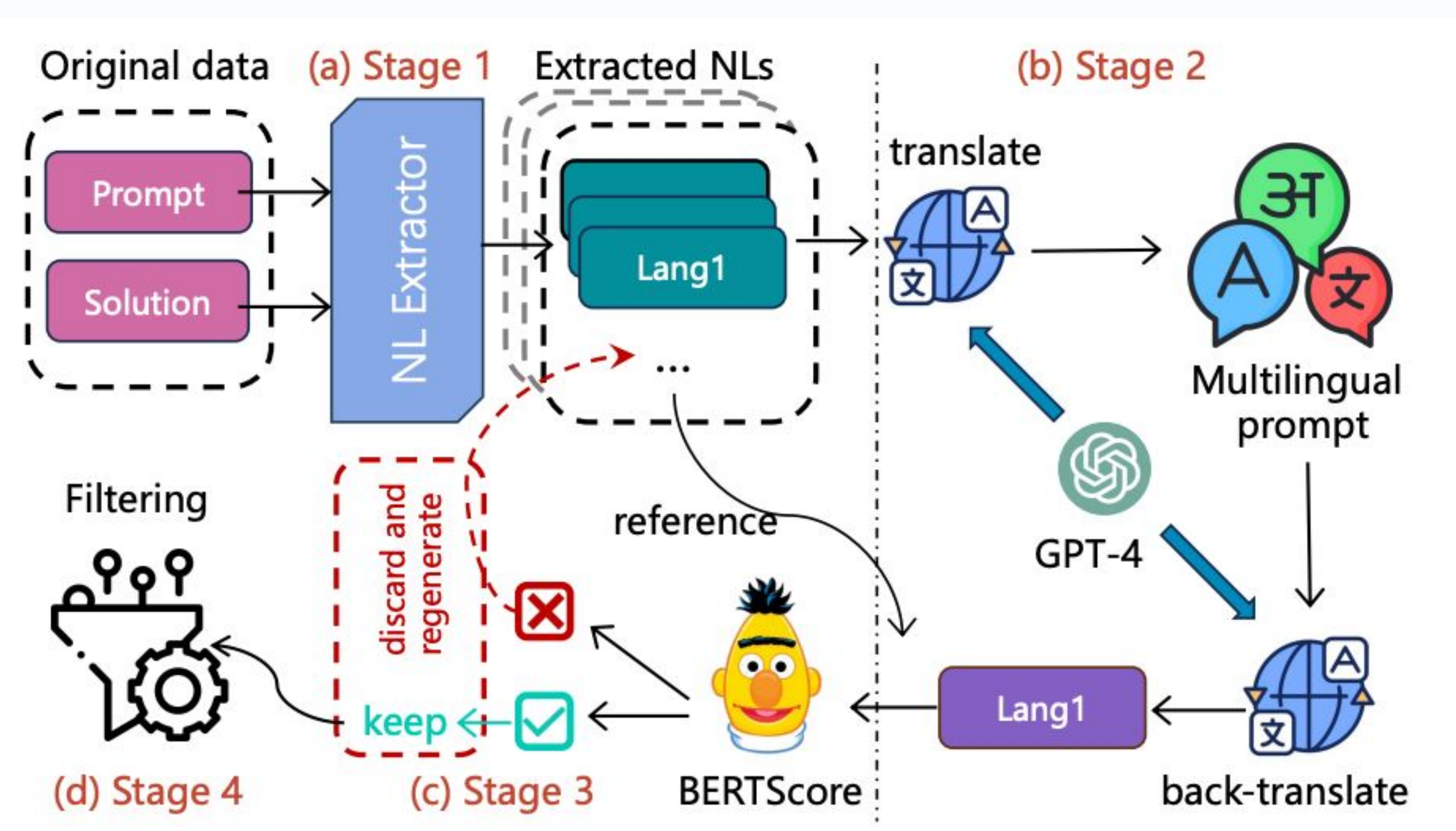
The task: can be formulated in different forms (e.g., code completion, variable/line infilling).

2 Motivation



- Current benchmarks primarily focus on **English** for code generation, limiting the relevant evaluation of LLMs on cross-lingual transfer.
- **High quality cross-lingual (NL) code generation benchmark** helps building better code generation models, leading to advanced code applications of **global impact and easy access for people from different regions**.

3 Dataset Construction



Construction Pipeline:

- **Text Extraction (Stage 1)**: We extract NL texts from the prompt.
- **Translation and Back-Translation (Stage 2)**: The extracted texts are translated into 23 different languages using GPT-4. These translations are then back-translated to English for subsequent automatic quality checks.
- **Quality Assessment with BERTScore (Stage 3)**: Stage 3 assesses translation quality by computing the BERTScore similarity score between the original English text and its back-translated text. Translations with a low similarity score (threshold < 0.95) are rejected and subjected to re-translation (max # of iter = 3).
- **Quality Control (Stage 4)**: Heuristic checks and manual evaluations are performed on the quality of the translated texts.

Dataset Statistics

Dataset	#Samples	#Average Test Cases	Data source	#PL	#NL	Parallel?
HumanEval (Chen et al., 2021)	164	7.7	Hand-written	1	1	✗
MBPP (Austin et al., 2021)	974	3.0	Hand-written	1	1	✗
APPS (Hendrycks et al., 2021)	10,000	13.2	Competitions	1	1	✗
DSP (Chandel et al., 2022)	1,119	2.1	Github Notebooks	1	1	✗
MTPB (Nijkamp et al., 2023b)	115	5.0	Hand-written	1	1	✗
DS-1000 (Lai et al., 2023)	1,000	1.6	StackOverflow	1	1	✗
Multlingual HumanEval (Athiwaratkun et al., 2023)	1,935	7.8	Hand-written	12	1	✗
ODEX (Wang et al., 2022)	945	1.8	StackOverflow	1	4	✗
HumanEval-XL	**22,080**	8.3	Hand-written	**12**	**23**	✔

Family	Languages
Afro-Asiatic	Arabic, Hebrew
Austro-Asiatic	Vietnamese
Austronesian	Indonesian, Malay, Tagalog
Indo-European (Germanic)	English, Dutch, German, Afrikaans
Indo-European (Romance)	Portuguese, Spanish, French, Italian
Indo-European (Greek)	Greek
Indo-European (Iranian)	Persian
Slavic	Russian, Bulgarian
Sino-Tibetan	Chinese
Turkic	Turkish
Uralic	Estonian, Finnish, Hungarian

The resulting **HumanEval-XL** consists of **80 parallel coding problems** spanning **12 PLs** and **23 NLs**. In total, this benchmark includes 22,080 coding problems.

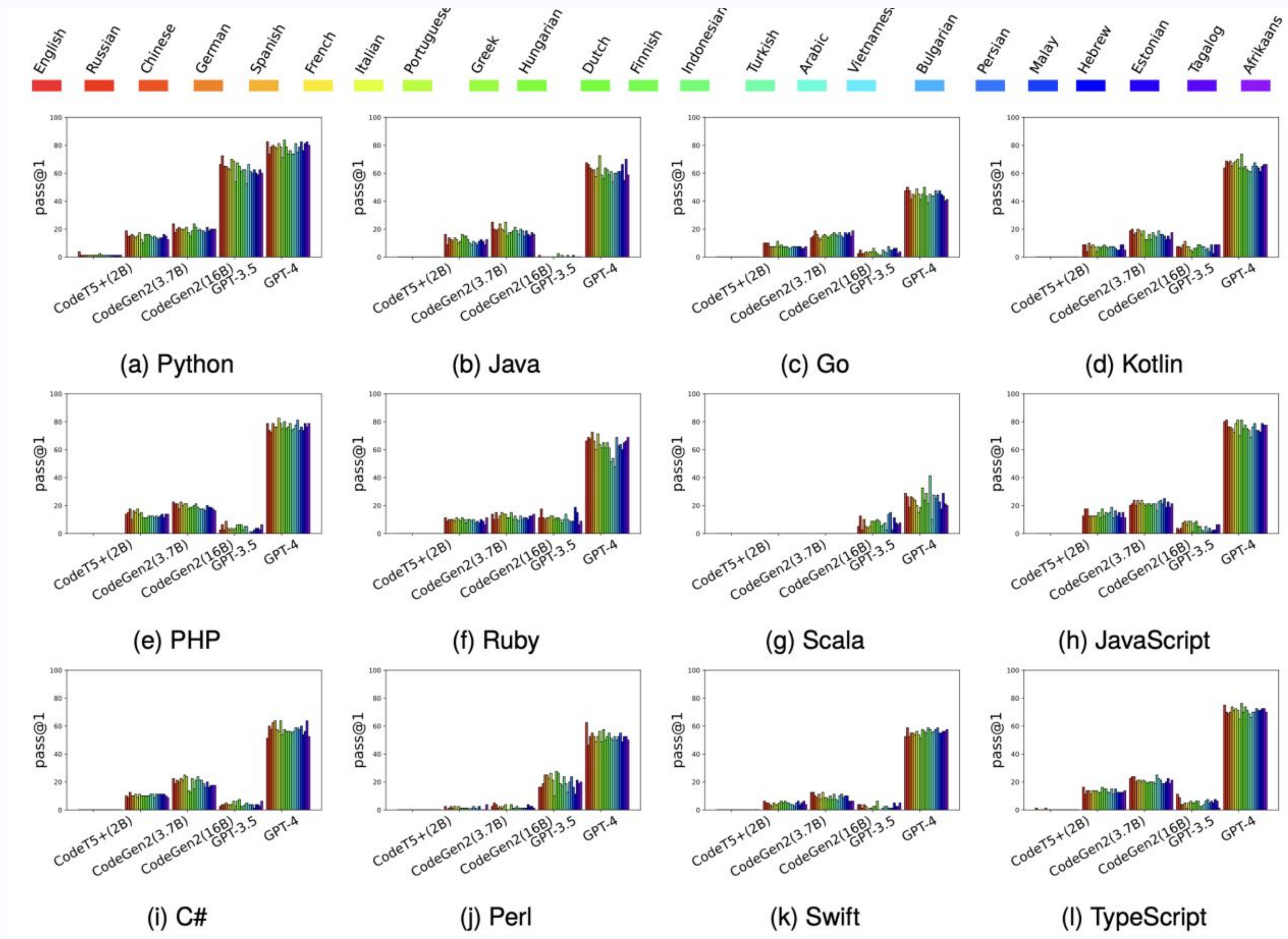
It further spans across **11 distinct language families**.

The **12 PLs** are the same as in Multilingual HumanEval, including Python, Java, Go, Kotlin, PHP, Ruby, Scala, JavaScript, C#, Perl, Swift and TypeScript. **23 NLs** are shown in the right figure.

4 Experiments

Main Results

We tested different models including **CodeT5+** (220M, 770M, 2B), **CodeGen2** (1B, 3.7B, 7B, 16B), **GPT-3.5**, and **GPT-4** on HumanEval-XL. Due to constrained computing resources, we report **pass@1** for all experiments (all experimental results can be found in the paper). We **order languages in their resource availability** as summarized in CC100 XL corpus.



(a) Python (b) Java (c) Go (d) Kotlin

(e) PHP (f) Ruby (g) Scala (h) JavaScript

(i) C# (j) Perl (k) Swift (l) TypeScript

Key Findings:

- Clear cross-lingual **inconsistency**.
- Increase in **model size** boosts performance.
- Specialized **code pre-training** plays a pivotal role in code generation.

Language Resource Analysis

Performance of different models on **Python** across grouped NLs. Average **pass@1** is reported

Model	Class 5	Class 4	Class 3
CodeT5+ (2B)	0.63±1.53	0.94±0.88	0.83±0.63
CodeGen2 (3.7B)	15.42±1.88	14.69±2.39	14.31±1.41
CodeGen2 (16B)	20.83±1.51	19.06±2.65	19.58±1.25
GPT-3.5	62.50±5.06	66.41±4.25	60.42±2.86
GPT-4	**78.54±2.90**	**78.75±3.54**	**77.64±4.07**

We have initially categorized the 23 NLs into **three distinct groups based on resource availability,** following the taxonomy outlined in Joshi et al. (2020) (ranging from 0 = least resourced to 5 = best resourced). **Class 5** contains EN, ES, FR, ZH, AR, DE. **Class 4** contains PT, IT, NL, RU, FI, VI, HU, FA. **Class 3** contains AF, ID, BG, EL, TL, MS, HE, ET, TR.

Language Family Analysis

Performance comparison of different models on **Python** across **language families**. Average **pass@1** is reported

Language Family	CodeT5+ (2B)	CodeGen2 (3.7B)	CodeGen2 (16B)	GPT-3.5	GPT-4
Afro-Asiatic	0.63±0.88	13.75±0.00	19.38±0.88	56.25±5.30	75.00±1.77
Austro-Asiatic	1.25±0.00	15.00±0.00	18.75±0.00	66.25±0.00	81.25±0.00
Austronesian	0.83±0.72	15.00±1.25	20.83±0.72	62.50±0.00	80.42±3.61
Indo-European (Germanic)	1.25±1.77	15.94±2.58	20.94±2.13	64.06±2.77	80.31±1.57
Indo-European (Romance)	0.31±0.62	15.31±1.57	20.31±0.63	66.25±3.68	79.06±1.57
Indo-European (Greek)	1.25±0.00	12.50±0.00	17.50±0.00	53.75±0.00	71.25±0.00
Indo-European (Iranian)	0.00±0.00	12.50±0.00	17.50±0.00	60.00±0.00	78.75±0.00
Slavic	0.63±0.88	14.38±0.88	18.13±0.88	66.88±7.95	74.38±0.88
Sino-Tibetan	0.00±0.00	15.00±0.00	20.00±0.00	65.00±0.00	78.75±0.00
Turkic	1.25±0.00	15.00±0.00	18.75±0.00	62.50±0.00	73.75±0.00
Uralic	1.25±1.25	14.17±3.61	19.58±4.39	62.50±4.51	79.58±5.20

We group languages into 11 distinct language families. The results underscore a significant challenge: Given NL prompts expressing the same meaning in different languages, current LLMs struggle to capture the equivalent semantic meaning.

5 Conclusion

- We propose **HumanEval-XL**, a massively multilingual code generation benchmark for assessing cross-lingual NL generation for LLMs.
- Our study reveals the **inconsistent cross-lingual transfer** of current LLMs (code/general), underscoring the significant challenge in achieving effective cross-lingual NL generalization.