Applied Computer Systems
ISSN 2255-8691 (online)
2026, vol. 31, no. 1, pp. 1–16
https://doi.org/10.2478/acss-2026-0001
https://content.sciendo.com

1862
RIGA TECHNICAL UNIVERSITY

Beyond Accuracy: Cross-Linguistic Equity and Socio-Technical Dimensions of Large Language Models

Fidan Kaya Gülağız*
Department of Computer Engineering, Kocaeli University, Kocaeli, Turkey

***Abstract* – Artificial intelligence (AI) and AI-based systems are rapidly gaining popularity across all areas of daily life. Among these systems, large language models (LLMs), which probabilistically model language to understand and generate text, stand out at the forefront. The ability to generate results from LLMs, whose primary focus is language, is of significant technical and social importance. As language diversity increases, the ability of LLMs to produce stable and consistent results is trending downwards. This decrease has a close relation with the size of the model, the scope of the training data, and the prompt technique used in response generation. To this end, a study was conducted to measure the success of LLMs in different languages. In the study, four LLMs were examined, three of which were open-source (DeepSeek-Coder-6.7B-Instruct, Qwen2.5-Coder-7B-Instruct, Llama-3.1-8B-Instruct) and one was closed-source (GPT-5). These models were evaluated using the HumanEval-XL dataset across seven natural languages that have different data sources and usage prevalences. Additionally, the effect of the human development index (HDI) values of the countries where the languages are spoken and the prompt technique used on the results was also analysed. Results show that as LLMs grow, performance differences between languages have decreased. Additionally, it has been observed that whether the models are open-source or closed-source also has a significant impact on performance. Among open-source LLMs, DeepSeek-Coder-6.7B-Instruct's accuracy rates range from 37 % to 60 %, while Qwen2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct have performed more consistently in the 95–99 % range. GPT-5, which is a closed-source LLM, has demonstrated balanced accuracy across all languages. The results obtained reveal remarkable results in ethics, quantity of linguistic data, and equality of access to technology. The results also clearly demonstrate the relationship between multilingual accuracy, language prevalence, and prompt techniques. In this way, the study offers a clearer and more comprehensive understanding of the issues surrounding linguistic justice and the generalization of LLMs in the field of AI.**

***Keywords* – Large language models, multilingual dataset, multilingual fairness, prompting techniques.**

I. Introduction

With the widespread use of the internet and advances in artificial intelligence (AI), an AI-focused transformation has begun in many aspects of our lives [1]–[3]. This process has influenced nearly every area of society, from modes of communication to knowledge generation, from education to health, and even to economic decisions. At the current stage, it is evident that the core of the process revolves around the concept of understanding humans, thinking, and making decisions like humans, as it does for AI itself. Consequently, the primary goal of this process is to develop systems capable of comprehending human language and acting accordingly. The most significant development in this domain in recent years has been the emergence of systems known as large language models (LLMs) [4]–[6]. LLMs can perform complex tasks in daily life by probabilistically modelling natural language [7], [8]. However, this rapid progress has also brought forth new ethical and societal concerns. Among these are issues such as linguistic injustice [9], the exclusion of low-resource languages [9], and inequalities in technological access [10].

Studies carried out in recent years [11]–[15] indicate that there are inequalities in the languages supported by LLMs. A study conducted by Kondoro [11] examined the use of AI writing assistants at Tanzanian universities. The study highlights that low-resource languages (especially Swahili) are underrepresented in AI systems. Findings show that students frequently use AI tools, but they do not benefit equally due to a lack of Swahili support, high costs, and poor infrastructure. The constitutional and legal dimensions of digital inequality and linguistic discrimination that emerged in the development of LLMs were examined by Ilin [15]. The study stated that languages with low levels of digitalization were underrepresented in AI systems. It argued that this restricted access to information, education and public services for communities speaking languages with low levels of digitization could constitute a violation of constitutional equality and human rights. Another study [12] examined the language-specific inequalities of LLMs in the process of acquiring new knowledge. Through experiments on 17 languages, the research showed that LLMs consistently suffered disadvantages in low-resource languages compared to high-resource languages in terms of effectiveness, transferability, prioritization, and robustness. A new metric was proposed by Li et al. [13] called "Language Ranker" to measure the performance differences of LLMs between high- and low-resource languages. The study

* Corresponding author's e-mail: fidan.kaya@kocaeli.edu.tr
Article received 2025-11-20; accepted 2026-02-02

showed that high-resource languages had more balanced language representations, whereas low-resource languages exhibited lower scores. In a different article addressing the inequalities faced by the African languages in the NLP field [14], it was stated that this situation is due to the connections between language policies, data availability, and model performance.

The main reason for the linguistic inequality problem encountered in LLMs is that LLMs are trained in languages that are commonly referred to as high-resource, and low-resource languages give relatively lower performance results than higher-resource languages due to their limited training data. This affects not only the accuracy of the LLM but also the level of technological utilisation by people in regions where low-resource languages are used. The conclusion here is that the less a society's language is represented, the less its members can benefit from LLM-based technologies [16]–[19]. This situation raises the problem of "linguistic injustice" [20] in access to technology.

There are many studies in the literature showing that prompt techniques also influence the accuracy of LLMs. Atreja et al. [21] examined how LLMs could be used in data labelling tasks in computational social sciences and how different prompt designs affected the accuracy and compliance of the model. In another study conducted by Chen [22], the quality of teaching was evaluated with a framework developed based on LLM, and the experimental results showed that well-designed prompts increased the quality of feedback received from LLM. Khojah et al. [23] investigated the effects of different prompt techniques on the function-level code generation performance of LLMs and, also, they introduced a dataset of 7072 requests designed for this purpose. Ma et al. [24] developed a new requirements-driven prompt engineering approach to guide people to use LLMs effectively. In the study conducted by Debnath et al. [25], the authors compared basic and advanced prompt techniques and analysed their strengths and weaknesses. In another study [26], LLMs were used as feedback tools in higher education. The authors examined the effect of the prompt quality used in this process. In a different review article [27], the role of prompt engineering techniques in maximising the potential of LLMs was extensively examined. The current studies reviewed show that the prompt technique used can affect the production accuracy of LLM. For this reason, the effect of the prompt technique should be considered in the analysis on the comprehensiveness of LLMs. In addition, recent findings [28] reveal that LLM performance is affected not only by linguistic representation and methodological choices but also by the human development index (HDI) of countries. In country-based evaluations, LLM accuracy is reported to be positively correlated with HDI [28].

This study mainly examines the cross-linguistic accuracy of different LLMs in coding problems. It also analyses the impact of the HDI of the countries where the selected languages are used, the digital content representation of the languages (in terms of the number of Wikipedia articles), and the prompt strategies (Zero-Shot Prompting (ZSP), Few-Shot Prompting (FSP), and Chain-of-Thought (CoT)) used on the accuracy of LLMs. For this purpose, four models were evaluated across seven different natural languages: three open-source models (DeepSeek-Coder-6.7B-Instruct, Qwen2.5-Coder-7B-Instruct, Llama-3.1-8B-Instruct) and one closed-source model (GPT-5). Considering the accessibility of the selected LLMs to different communities, open-source LLMs were preferred. The main contributions of the study are listed below.

- The study adds a social dimension to the evaluation process of LLMs and offers comparisons in terms of HDI, number of language speakers and linguistic resource richness.
- The study evaluates the effect of three prompt techniques on LLM accuracy through code generation problems and quantitatively reveals the role of LLMs in this field.
- By including both open and closed-source LLMs in the comparison, it analyses the differences in performance, access and fairness between open- and closed-source LLMs in a holistic problem-specific manner.
- In addition, it is one of the pioneering studies that systematically analyses the relationship between LLM performance and countries' HDI, number of resources and number of speakers.

The remainder of this study is organised as follows: Section II details the dataset, language and LLM selection process, prompt techniques and evaluation metrics. Section III presents the experiments and results. Section IV provides an extended discussion that includes prompt technique effect, cross-linguistic performance of LLMS, effect of HDI and linguistic factors, open- and closed-source LLM performance differences, LLM language prioritization differences and limitations of the study. Section V summarises the key findings and highlights the contributions.

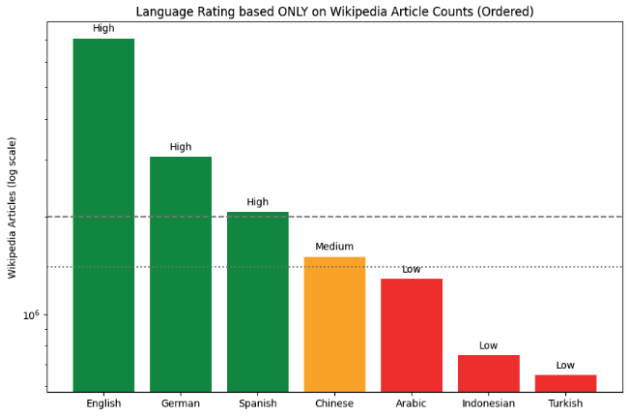
II. Material and Methods

A. Multilingual Experimental Dataset and Language Selection

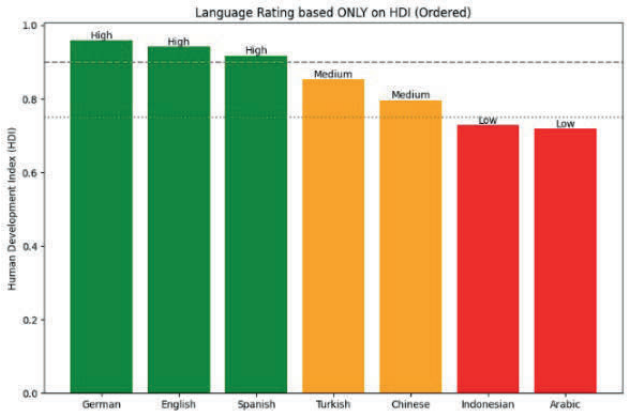
The HumanEval-XL dataset [29] was used to measure the code generation and reasoning abilities of the LLMs selected for the study. This dataset represents a multilingual subset of the HumanEval dataset [30] created in 2021 with 164 questions specific to a single programming language and a single language. The HumanEval-XL dataset, introduced in 2024, contains 80 parallel problems developed across 12 different programming languages and 23 different natural languages [29]. Seven natural languages were selected from this dataset. These languages were Indonesian, Arabic, Turkish, Spanish, Chinese, German, and English. (The term "natural languages" refers to the language in which the problems to be solved are expressed, and Python refers to the programming language in which the solution code must be written). The languages were chosen to represent different language families and global regions. Three different indicators, specifically chosen for the study, were used to categorise the selected languages as low, medium and high.

TABLE I
LANGUAGE RESOURCE EVALUATION BASED ON WIKIPEDIA COVERAGE, HDI, AND COMPOSITE SCORE

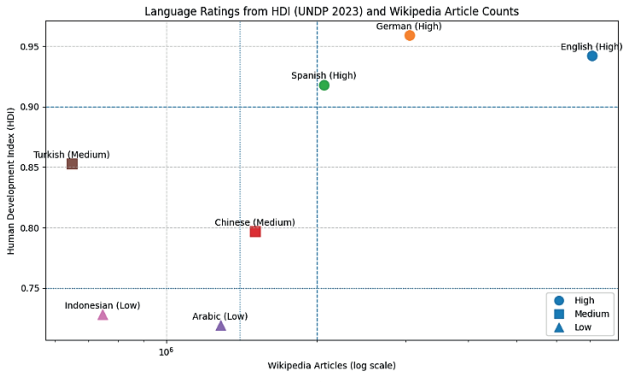
Language	Wikipedia Articles Count [31]	HDI [32]	Composite Score	Rating
English	7 068 119	0.942	2.905 118	High
German	3 056 078	0.959	1.995 713	High
Spanish	2 065 147	0.918	1.040 464	High
Chinese	1 503 305	0.797	–0.684 265	Medium
Turkish	648 121	0.853	–1.175 005	Medium
Arabic	1 282 217	0.719	–1.736 579	Low
Indonesian	746 024	0.728	–2.345 446	Low



(a)



(b)



(c)

Fig. 1. Comparative language ratings based on (a) Wikipedia article counts, (b) HDI, and (c) composite score.

1. Wikipedia Article Count [31]: It expresses the online encyclopaedic content produced in a language. This indicator shows relatively which languages can have richer data sources in the education of LLMs.
2. HDI [32]: It expresses the level of human development of a country or language community based on education, health, and income metrics. Languages with high HDI values are generally considered to have strong research bases.
3. Composite Score: Obtained by combining Wikipedia article count and HDI values. Thus, instead of focusing solely on the amount of digital content or social sophistication, a more balanced rating metric that takes both dimensions together is presented.

The overall score was calculated by averaging two normalized indicators (Wikipedia article count and HDI) for each language ((1) and (2)). In the first step, the Wikipedia article count and HDI indicators were normalized to a range of 0–1 according to (1):

$$Norm(x) = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{1}$$

$$Score(x) = \frac{Norm(Wikipedia) + Norm(HDI)}{2}. \tag{2}$$

In (1), the x variable represents the actual value of the indicator to be normalized for the relevant language, min(x) represents the lowest value of the relevant indicator in Table I, and max(x) represents the highest value of the relevant indicator in Table I. Table I uses the number of Wikipedia articles for languages as of 3 October 2025, and HDI data from the UNDP Human Development Report [32]. HDI measures the human development level of countries. As some languages are spoken in more than one country, the development index data is included in the table as follows, in the language-based analysis:

- For English, as in common academic practice, the United Kingdom was taken as the reference country (HDI value: 0.946).
- For Arabic, instead of a single country, the regional average for the “Arab States” reported by the UNDP was used (HDI value: 0.719). This approach provides a more

representative measurement by taking into account the multinational nature of Arabic.

The reference countries for English and Arabic were chosen to increase comparability and more accurately reflect the global distribution of languages. Alternatively, using the United States for English or a high-HDI country like Qatar for Arabic was considered; however, the UK provides a traditional academic reference point, and the regional average for Arabic prevents the composite score from being overly influenced by the specific economic conditions of a single nation. This approach aims to minimise geographical bias and ensures that the HDI component reflects a broader linguistic community rather than a localized one. According to the categorisation based on the composite score mentioned above, English and German are in the "High" category thanks to both high digital content and high HDI, while Turkish and Chinese are classified in the "Medium" category as languages with medium performance. The Indonesian language is attributed to the "Low" category due to its low Wikipedia content and relatively low HDI value. The objective of equally weighting Wikipedia article count and HDI in the composite score is to ensure a balanced representation of both digital presence and development. This aims to distinguish between languages spoken by a large number of people but with limited digital data, and languages that possess the rich resources required for AI models to learn and the socio-economic infrastructure to develop this technology. Figure 1 visualises comparative language ratings respectively based on Wikipedia article counts, HDI, and composite score values.

The technique used to categorise languages in this study and the classifications of "high," "medium," and "low" obtained through the composite score are not universal or standard. These categories were specifically defined for this research, based on relative thresholds obtained from two preferred criteria within the scope of the study (number of Wikipedia articles and HDI 2023 values). Therefore, the presented classification provides a study-specific analytical framework. Different studies may yield different categorisations based on the different metrics chosen.

B. LLMs for Experimental Evaluation

The study evaluates the performance of popular open-source LLMs in solving code generation problems. For this purpose, open-source LLMs, mostly optimised for coding tasks and with a scale of about 7–8 billion parameters, were selected. In addition to high performance, LLMs at this parameter scale can be run in widely preferred development environments (e.g., Colab Pro, where LLMs can run comfortably on Colab Pro's 16–24 GB VRAM hardware with preferred parameter scales). Thus, different publicly accessible open-source LLMs could be compared relatively fairly at similar capacity levels. In addition, both Western and Eastern models are included in the study, considering the diversity of LLM developers. However, to show the current position of open-source models compared to the current, closed-source LLMs, GPT-5, the latest generation model developed by OpenAI, is also included in the study as a reference point. For this purpose, the models included in the study were DeepSeek-Coder-6.7B-Instruct [33], Qwen2.5-Coder-7B-Instruct [34], Llama-3.1-8B-Instruct [35], and GPT-5 [36].

Table II provides a comprehensive summary of the developer, prompt style, parameters, and primary language focus of the LLMs included in the study. The temperature parameter is set to 0.0 for all LLMs. The aim is to run LLMs with greedy decoding logic and maximise the reproducibility of the results. This setting is recommended in the official documentation of the DeepSeek API [37]. Additionally, recent studies show that low or zero temperature values reduce randomness and produce clearer outputs for code-based problems [38], [39]. The number of outputs was set to 192 to focus on the ability of the models to generate concise but accurate solutions that align with real-world applications. Thus, each model's robustness and reasoning ability under a minimum output budget were analysed. Generation settings were kept consistent across models for stability rather than reflecting default values.

TABLE II
EVALUATION SETTINGS OF LLMS

Model	Developer	Prompt Style	Parameters	Primary Language Focus
DeepSeek-Coder-6.7B-Instruct	DeepSeek AI	DeepSeek	temperature=0.0, max_new_tokens=192, n_ctx=4096, n_gpu_layers=-1, n_batch=1024	Chinese + English [40]–[42]
Qwen2.5-Coder-7B-Instruct	Alibaba Cloud	ChatML		Chinese + English [43]
Llama-3.1-8B-Instruct	Meta AI	Llama-3		English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai [44]
Gpt-5	OpenAI	default	temperature=0.0, other parameters=default	–

DeepSeek-Coder-6.7B-Instruct [33] is a model trained from scratch on approximately two trillion tokens, with 87 % of its data consisting of code and 13 % being the English and Chinese natural language [40]. The model is optimised for code generation across multiple programming languages (Python, Java, C++, Go, etc.) and has an instruction-tuned design [40]. Its developers have released quantified versions to ensure usability in resource-limited environments, while still delivering a performance competitive with larger closed-source models [38].

Qwen2.5-Coder-7B-Instruct [34] is a member of the Qwen2.5 family, developed by Alibaba Cloud Intelligence and specifically designed for code generation tasks [34]. The model has been developed with a focus on the Chinese and English languages [43]. The Qwen2.5-Coder-7B-Instruct model has been reported to perform particularly well in coding tasks and produce results comparable to larger models [34].

Llama-3.1-8B-Instruct [35] is one of the latest models of Meta AI to be developed [35]. Released as part of Meta's open-source strategy, the Llama-3 series is described by Meta as "made accessible for individuals, creators, researchers, and businesses of all sizes to experiment, innovate, and scale their ideas" [45]. The main languages officially supported by the model are English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai [44].

All open-source models were evaluated locally using llama.cpp (cuBLAS GPU) and GGUF weights. All runs were executed on A100-40GB (CUDA 12.x) under the following settings: Q5_K_M quantization priority (with Q5_0/Q4_K_M/Q4_0 fallbacks when necessary), n_ctx = 4096, temperature = 0.0, max_new_tokens = 192, NUM_SAMPLES = 10. Outputs were verified through unit testing and pass@1 / pass@k measurements were reported. Model-specific prompt templates were used. In all cases, only output code was requested, and description/comment lines were disabled.

In addition to open-source LLMs, the closed-source LLM GPT-5 developed by OpenAI was also included in the study. It represents the latest generation of LLMs in the GPT series. GPT-5 offers enhanced AI performance in a wide range of domains, from coding to mathematics, text generation, healthcare, and visual perception. The model has an architecture that allows it to decide between generating quick responses and engaging in long-term reasoning [46]. It is available in three versions: Basic, Plus, and Pro. The GPT-5 model can also operate in three main modes: Instant Thinking, Auto Thinking, and Thinking, which differ in the reasoning they employ [46]. All experiments using GPT-5 were conducted via the OpenAI API. The model identifier was set to "gpt-5" that corresponds to the standard version of GPT-5 available on the OpenAI API as of 2025. All results reflect the basic reasoning and code generation capabilities of the standard GPT-5 model.

C. Prompt Techniques

In this study, not only linguistic and model-related differences but also the effects of prompting techniques were systematically examined. ZSP [47], FSP [47], and CoT [48] based prompts were generated for the same task set; each technique was run under the same execution parameters (context window, temperature, top_p, max_new_tokens, etc.) and the same evaluation protocol (unit tests, pass@1 / pass@k). Templates were constrained to return "the final solution code only", and the task order was fixed to reduce potential biases. At the same time, the use of only Python language was required to solve coding problems in the dataset.

TABLE III
PROMPT TECHNIQUES AND CORE TEMPLATES USED IN THIS STUDY

Technique	Purpose	Prompt skeleton (key lines)
ZSP	Solve the task from instructions without examples.	• You are a Python assistant. • Return only the full Python implementation. • No explanations or comments. • No print, input, file I/O, or Internet access.
FSP	Teach label/format and patterns via a few demonstrations.	• Same code-only rules plus compact demos: • Example Problem–Solution #1/2 … • Solve the following problem:
CoT	Encourage step-by-step internal reasoning before answering.	• Same code-only rules; additionally: • Think step-by-step, • Provide only the final Python code, and a • (Internal) Hidden Thinking Process checklist.

The two prompting techniques preferred in this study are ZSP and FSP. The methods are based on the In-Context Learning (ICL) paradigm. ICL is a general paradigm that allows LLMs to adapt to new tasks without training relying solely on instructions inserted into the context and a few optional examples [49]. ZSP and FSP are special case versions of the ICL paradigm. When the ZSP technique is used to generate prompts for LLMs, the models perform a task using instructions alone, without examples [25], whereas in the FSP technique, a small number of examples are included within the instructions to execute the task [47]. ICL provides an interpretable interface, allowing human knowledge to be easily integrated into the model. It operates similarly to analogy-based decision-making and reduces the computational cost associated with adapting to new tasks with its "training-free" structure [49]. As specific implementations of this paradigm, ZSP and FSP [50] inherit the same advantages. In the literature, ZSP and FSP approaches have shown effective performance across a wide range of NLP tasks, including machine translation, question answering, and code generation [47]. One of these areas is code generation. For instance, Chen et al. [30] evaluated zero-shot Python program synthesis on HumanEval using the pass@k metric in the Codex study and reported strong results, while Austin et al. [51] demonstrated the effectiveness of few-shot prompts on the MBPP dataset consisting of programming tasks. Consequently,

both ZSP and FSP prompt techniques were included in the HumanEval-XL evaluation for problem solving in this study.

The third prompting technique used in this study is CoT. This technique is a specific application of the ICL paradigm [49]. It is like an enriched version of ICL with intermediate reasoning steps added to the context [49]. The CoT technique improves logical consistency by guiding the model to rationalize the solution step by step [25], [48]. With this ability, performance significantly improves on high-complexity tasks such as reasoning, logical inference, and multi-step Q&A [25], [52]. CoT enhances generalization, even with small examples (zero-shot), enabling models to better adapt to new tasks [25], [53]. In addition to all these advantages, it requires more processing and time costs because it provides additional reasoning [54].

Table III presents the purposes and core templates of the three prompting techniques used in this study. All prompts enforced code-only outputs; this rule is stated explicitly in ZSP/FSP and reinforced by the system message and rules in CoT; also, it is clearly stated in the prompts that the codes will be written in the Python language. As can be seen from the table, the CoT template, unlike ZSP and FSP, includes a "think step-by-step" and an (internal) "Hidden Thinking Process" checklist. To ensure a fair cross-linguistic evaluation, the prompt templates were translated verbatim from the English baseline into each natural language without any stylistic adaptation. Additionally, no language-specific tuning, few-shot selection, or hyperparameter optimisation was performed for individual languages; all models were evaluated under identical settings to measure their out-of-the-box multilingual performance.

D. Evaluation Metrics

In this study, accuracy was used as the primary metric for evaluating LLMs; the output of each problem was labelled as correct/incorrect according to whether it passed the relevant unit tests, and the proportion of correct answers was reported. Accuracy is calculated as shown in (3). The N value in the formula represents the number of problems in the dataset, and $y_{i,1} \in \{0,1\}$ represents whether the first generated code sample for the i-th problem passes the tests.

$$Accuracy_{(1\ \text{sample})} = \frac{1}{N}\sum_{i=1}^{N} y_{i,1} \tag{3}$$

As a complement to Accuracy, pass@1 and pass@k, two metrics commonly used for evaluation in the code generation literature, are also reported. Theoretically, pass@1 represents the probability that a single sample will pass the tests for a task, and pass@k represents the probability that at least one of the k independent/identical samples will pass the tests for the same task. In the multi-sample case, it is necessary to use unbiased estimators for these metrics [30]. The relevant unbiased estimators are given in (4) and (5). Here, n_i represents the number of samples generated for the i^{th} problem and c_i represents the number of samples that passed the tests.

$$\widehat{pass@1} = \frac{1}{N}\sum_{i=1}^{N} \frac{c_i}{n_i}, \tag{4}$$

$$\widehat{pass@k} = \frac{1}{N}\sum_{i=1}^{N} \left(1 - \frac{\binom{n_i - c_i}{k}}{\binom{n_i}{k}} \right), \; k = \min\left(K, n_i\right). \tag{5}$$

In the study, the output is deterministic since the code generation of LLMs is carried out with the temperature = 0 (greedy) condition [55], [56]. In the single sample case, $Accuracy_{(1\ \text{sample})}$ is practically identical to pass@1. In the multi-sample case, when the generated trials give the same output, pass@k is also reduced to this value; however, for some problems, when $0 < c_i < n_i$ (some trials pass and some do not in the same task) is observed, $Accuracy_{(1\ \text{sample})}$ (3) and unbiased pass@1 (4) and unbiased pass@k (5) may differ. These differences are mostly due to timeouts, environment volatility or logging gaps in the test environment [57]. The primary metric in the study is $Accuracy_{(1\ \text{sample})}$, but to ensure comparability, unbiased metrics (pass@1 and pass@k) that weight all problems equally are also reported.

III. Experimental Study

The section presents the experimental study conducted to evaluate the performance of four major language models across seven different natural languages (English, German, Spanish, Chinese, Arabic, Turkish, and Indonesian). The experiments were conducted using the multilingual version of the HumanEval-XL dataset. Three different prompting techniques were applied to the dataset to test the models' instruction-following capabilities, knowledge transfer capabilities, and cross-lingual consistency. This evaluation aims to reveal the effects of language coverage, the model's training language priorities, and prompting strategies on multilingual code generation accuracy and cross-lingual generalization.

Table IV shows the $Accuracy_{(1\ \text{sample})}$ values of LLMs on the HumanEval-XL dataset with seven natural languages and three different prompting techniques. Figure 2 shows the comparative accuracy graphs by prompting type for all LLMs included in the study.

According to Table IV, GPT-5 showed accuracy in the 95–100 % range and for CoT technique showed consistent accuracy (100 %) in all languages, the Qwen2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct models also exhibited near-maximal accuracy in the 95–98.75 % range. In contrast, DeepSeek-Coder-6.7B-Instruct's performance varies widely across languages, with the highest results in Chinese and Arabic (about 58.75 % under ZSP for Chinese and about 58.75 % under FSP for Arabic) and the lowest results in German, depending on the prompt technique.

TABLE IV

ACCURACY$_{(1\ SAMPLE)}$ RESULTS OF FOUR LLMS ACCORDING TO SEVEN NATURAL LANGUAGES AND THREE PROMPTING TECHNIQUES (ZSP, FSP, AND COT)

LLM	**Prompt Technique**	**Languages**						
		English	**German**	**Spanish**	**Chinese**	**Turkish**	**Arabic**	**Indonesian**
DeepSeek-Coder-6.7B-Instruct	ZSP	35	46.25	42.5	58.75	52.5	55	40
	FSP	30	15	37.5	31.25	32.5	58.75	30
	CoT	36.25	15	33.75	47.5	52.5	41.25	35
Qwen2.5-Coder-7B-Instruct	ZSP	98.75	97.5	97.5	97.5	96.25	98.75	97.5
	FSP	98.75	95	95	96.25	97.5	96.25	95
	CoT	98.75	95	98.75	98.75	96.25	97.5	97.5
Llama-3.1-8B-Instruct	ZSP	98.75	96.25	96.25	95	92.5	95	98.75
	FSP	95	96.25	97.5	96.25	98.75	96.25	97.5
	CoT	98.75	98.75	97.5	98.75	97.5	96.25	98.75
GPT-5	ZSP	98.75	97.5	100	98.75	96.25	97.5	97.5
	FSP	96.25	96.25	98.75	95	95	98.75	100
	CoT	100	100	100	100	100	100	100

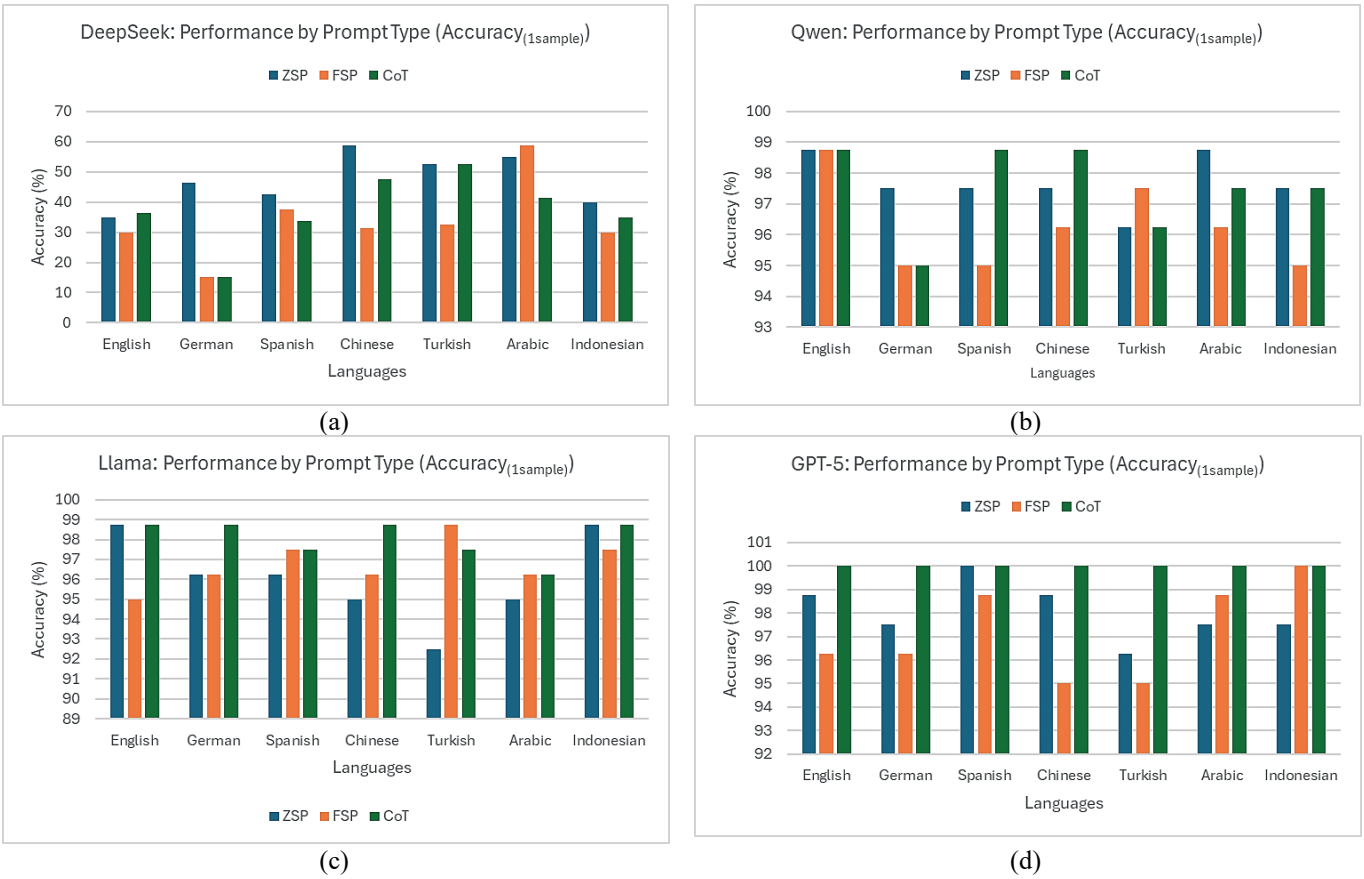


Fig. 2. Comparative visualisation of Accuracy$_{(1\ sample)}$ results across prompting techniques (ZSP, FSP, CoT) for four LLMs: (a) DeepSeek-Coder-6.7B-Instruct, (b) Qwen2.5-Coder-7B-Instruct, (c) Llama-3.1-8B-Instruct, and (d) GPT-5.

Table V presents the unbiased Pass@1 and Pass@10 accuracy scores obtained by DeepSeek-Coder-6.7B-Instruct across seven natural languages under three prompting strategies: ZSP, FSP, and CoT. The numerical results summarise performance consistency within each language and prompting conditions without interpretive analysis.

TABLE V
UNBIASED PASS@1 AND PASS@10 ACCURACY RESULTS OF DEEPSEEK-CODER-6.7B-INSTRUCT ACROSS SEVEN NATURAL LANGUAGES UNDER THREE PROMPTING STRATEGIES (ZSP, FSP, AND COT)

	DeepSeek-Coder-6.7B-Instruct					
	ZSP		**FSP**		**CoT**	
Languages	Pass@1	Pass@10	Pass@1	Pass@10	Pass@1	Pass@10
English	37.25	40	27.75	31.25	32.88	38.75
German	44	47.5	18.38	18.75	18.38	18.75
Spanish	41.38	46.25	34.12	40	32.63	36.25
Chinese	57.62	60	26.75	33.75	43	50
Turkish	50.25	52.5	34.75	38.75	52.5	56.25
Arabic	55	58.75	57.63	60	41.25	42.5
Indonesian	38.8	45	26.62	30	32.75	36.25

Figure 3 (a) and (b) illustrate the same results in heatmap form, depicting the variation of Pass@1 and Pass@10 values across languages and prompt types. Overall, Pass@10 values are consistently higher than Pass@1, indicating that the model is able to generate at least one correct solution within its top 10 outputs more frequently than in its first prediction. The average gain between Pass@1 and Pass@10 ranges from 0 to +7 percentage points, depending on language and prompt type. Under ZSP, the model reaches its highest accuracies in Chinese (57.62 → 60), Arabic (55 → 58.75), and Turkish (50.25 → 52.5), while the lowest results appear in Indonesian (38.8 → 45) and English (37.25 → 40). FSP produces markedly lower scores in most languages, particularly for German (18.38 → 18.75) and Indonesian (26.62 → 30), with the only relatively strong outcome in Arabic (57.63 → 60). Under CoT prompting, performance shows a slight improvement only in Turkish (52.5 → 56.25 for Pass@10), while results in other languages are comparable to or lower than those achieved with ZSP. Figure 3 visualises these results, where lighter tones in the heatmaps represent higher accuracy values for both Pass@1 and Pass@10.

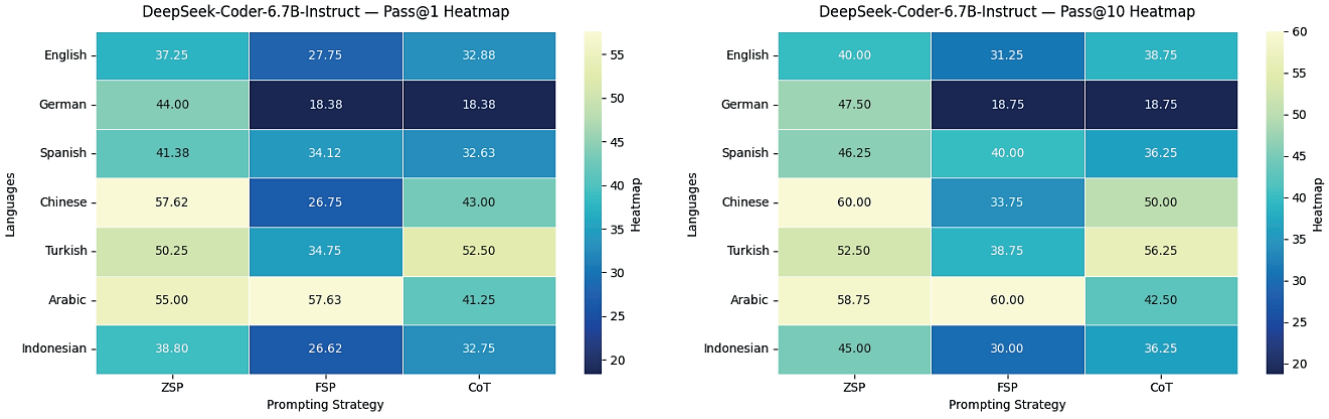


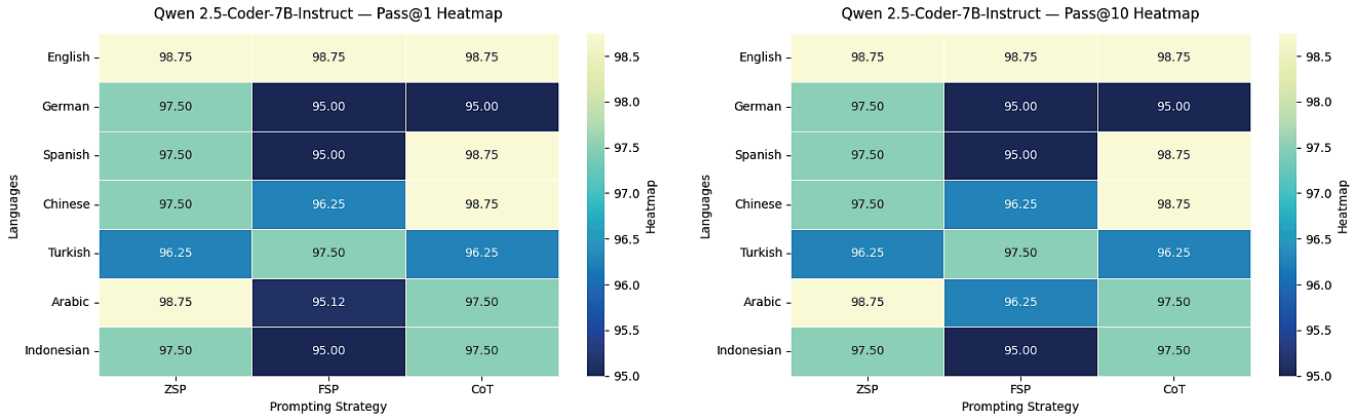
Fig. 3. Heatmap visualisation of DeepSeek-Coder-6.7B-Instruct results: (a) Pass@1 and (b) Pass@10 accuracies across seven natural languages and three prompting strategies (ZSP, FSP, CoT).

Table VI presents the unbiased Pass@1 and Pass@10 accuracies of Qwen2.5-Coder-7B-Instruct across seven natural languages using three prompting strategies. Overall, both Pass@1 and Pass@10 values remain nearly identical (ranging between 95 % and 98.75 %), indicating that the model's first prediction is almost always correct, and additional sampling provides negligible improvement. Under ZSP, the model often achieved higher accuracy rates than other prompting techniques, reaching 98.75 % for English and Arabic, 97.5 % for German, Spanish, Chinese, and Indonesian, and 96.25 % for Turkish. FSP and CoT produce comparably strong results, with only minor deviations (±0–4 points) from the ZSP baseline. In terms of Pass@1 and Pass@10 metrics, when the model is evaluated on a language basis, except Arabic, both metrics converged to the same upper bound performance. When a comparison is made between languages, Qwen2.5-Coder-7B-Instruct achieved the highest accuracy for both metrics for English, with 98.75 for all prompt techniques. Figure 4 presents visualised heatmaps of the results given in Table VI. Lighter shades in the heatmaps represent higher accuracy values for the Pass@1 and Pass@10 metrics.

TABLE VI

UNBIASED PASS@1 AND PASS@10 ACCURACY RESULTS OF QWEN2.5-CODER-7B-INSTRUCT ACCORDING TO SEVEN NATURAL LANGUAGES AND THREE PROMPTING TECHNIQUES (ZSP, FSP, AND COT)

	Qwen-2.5-Coder-7B-Instruct					
	ZSP		**FSP**		**CoT**	
Languages	Pass@1	Pass@10	Pass@1	Pass@10	Pass@1	Pass@10
English	98.75	98.75	98.75	98.75	98.75	98.75
German	97.5	97.5	95	95	95	95
Spanish	97.5	97.5	95	95	98.75	98.75
Chinese	97.5	97.5	96.25	96.25	98.75	98.75
Turkish	96.25	96.25	97.5	97.5	96.25	96.25
Arabic	98.75	98.75	95.12	96.25	97.5	97.5
Indonesian	97.5	97.5	95	95	97.5	97.5



(a) (b)

Fig. 4. Heatmap visualisation of Qwen2.5-Coder-7B-Instruct results: (a) Pass@1 and (b) Pass@10 accuracies across seven natural languages and three prompting strategies (ZSP, FSP, CoT).

Table VII presents the unbiased Pass@1 and Pass@10 accuracies of Llama-3.1-8B-Instruct across seven natural languages using three prompting strategies. The results show that accuracy values across all languages and prompt types ranged from approximately 93 % to 100 %.

TABLE VII

UNBIASED PASS@1 AND PASS@10 ACCURACY RESULTS OF LLAMA-3.1-8B-INSTRUCT ACROSS SEVEN NATURAL LANGUAGES UNDER THREE PROMPTING STRATEGIES (ZSP, FSP, AND COT)

	Llama-3.1-8B-Instruct					
	ZSP		**FSP**		**CoT**	
Languages	Pass@1	Pass@10	Pass@1	Pass@10	Pass@1	Pass@10
English	98.75	98.75	97.25	97.5	98.75	98.75
German	96.25	96.25	95.12	96.25	98.75	98.75
Spanish	95.12	96.25	97.5	97.5	97.5	97.5
Chinese	95	95	96.25	96.25	98.75	98.75
Turkish	93.62	93.75	98.75	98.75	96.3	97.5
Arabic	95	95	96.25	96.25	96.25	96.25
Indonesian	99.88	100	97.5	97.5	98.75	98.75

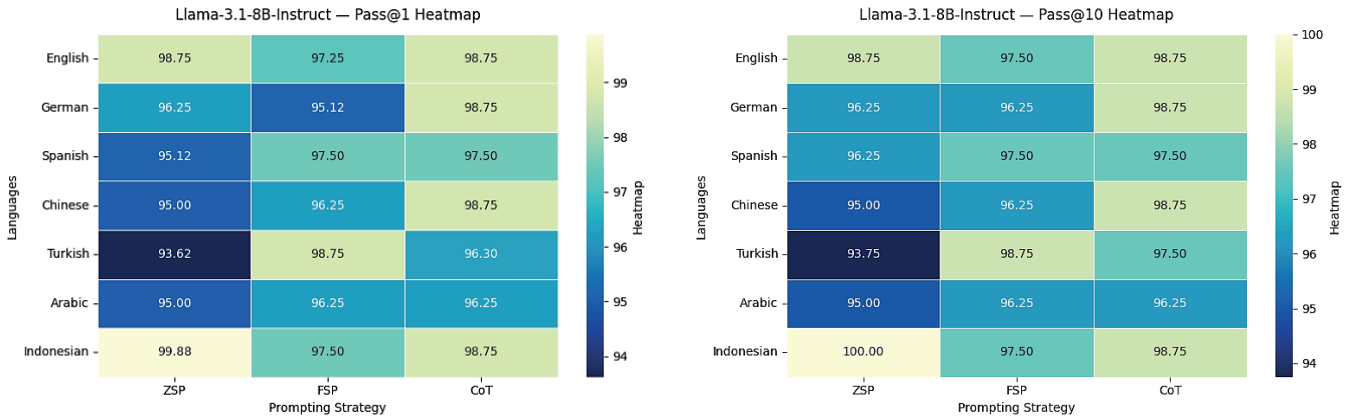


Fig. 5. Heatmap visualisation of Llama-3.1-8B-Instruct results: (a) Pass@1 and (b) Pass@10 accuracies across seven natural languages and three prompting strategies (ZSP, FSP, CoT).

The differences between the Pass@1 and Pass@10 metrics are small, usually in the range of about ±0–2 percentage points. This shows the potential of Llama-3.1-8B-Instruct to produce correct outputs on the first try in code writing problems. When ZSP was used as the prompt technique, the accuracy remained above 93 % in all languages, with the highest accuracy of 99.88 → 100 in Indonesian and 98.75 → 98.75 in English. Similarly strong results were obtained with the FSP technique. Accuracy remained above 95 % in all languages. The FSP technique showed slight accuracy improvements over the ZSP technique, especially in Spanish, Turkish, Chinese and Arabic, but produced less accurate codes in English, German and Indonesian. CoT prompting techniques generally outperformed other prompting techniques with accuracy rates between 96 % and 99 % in all languages. Figure 5 presents visualised heatmaps of the results given in Table VII. Lighter shades in the heatmaps represent higher accuracy values for the Pass@1 and Pass@10 metrics and visually confirm the overall stability of the model and minimal variation between prompting strategies.

Table VIII presents the unbiased Pass@1 and Pass@10 accuracies for three open-source LLMs using their best-performing prompting strategies: ZSP for DeepSeek-Coder-6.7B-Instruct and CoT for both Qwen-2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct. As shown in Table VIII, DeepSeek-Coder-6.7B-Instruct achieves moderate accuracies, with Pass@1 values between 37.25 % and 57.62 %, and Pass@10 between 40 % and 60 %. In contrast, Qwen2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct maintain consistently high accuracies above 95 % for both metrics across all languages, showing minimal differences between Pass@1 and Pass@10. Among high-performing models, Qwen2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct display nearly identical cross-lingual stability, with Qwen2.5-Coder-7B-Instruct reaching its lowest accuracy in German (95 %) and Llama-3.1-8B-Instruct showing uniformly high results across all languages. Figure 6 visualises these outcomes, where the bars corresponding to Qwen2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct remain close to the upper accuracy limit, while DeepSeek-Coder-6.7B-Instruct exhibits larger variation among languages. Collectively, these results provide a comparative overview of how open-source models differ in multilingual accuracy under their most effective prompting configurations.

TABLE VIII

UNBIASED PASS@1 AND PASS@10 ACCURACY RESULTS OF THE BEST-PERFORMING PROMPTING TECHNIQUES FOR OPEN-SOURCE LLMS ACROSS SEVEN NATURAL LANGUAGES

LLM	**Metric**	**Languages**						
		English	**German**	**Spanish**	**Chinese**	**Turkish**	**Arabic**	**Indonesian**
DeepSeek-Coder-6.7B-Instruct (ZSP)	Pass@1	37.25	44	41.38	57.62	52.5	55	38.8
	Pass@10	40	47.5	46.25	60	52.5	58.75	45
Qwen2.5-Coder-7B-Instruct (CoT)	Pass@1	98.75	95	98.75	98.75	96.25	97.5	97.5
	Pass@10	98.75	95	98.75	98.75	96.25	97.5	97.5
Llama-3.1-8B-Instruct (CoT)	Pass@1	98.75	98.75	97.5	98.75	96.3	96.25	98.75
	Pass@10	98.75	98.75	97.5	98.75	97.5	96.25	98.75

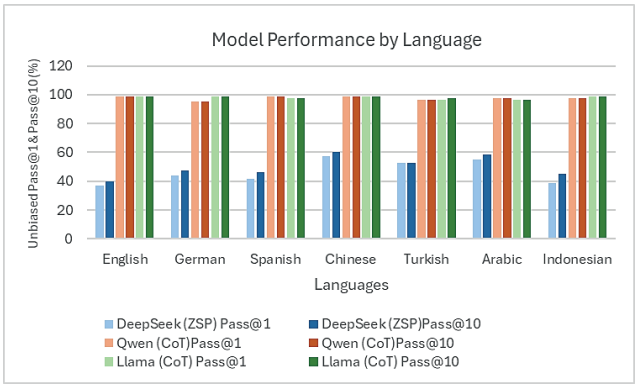


Fig. 6. Comparative visualisation of the best-performing prompting techniques for open-source LLMs showing unbiased Pass@1 and Pass@10 accuracies across seven natural languages.

Figure 7 visualises the relationship between HDI and model performance ($\text{Accuracy}_{(1\text{ sample})}$) across four LLMs. In this analysis, the Accuracy values correspond to the best-performing prompting technique for each model: ZSP for DeepSeek-Coder-6.7B-Instruct, CoT for Qwen2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct, and constant accuracy values across all strategies for GPT-5. The correlation coefficient values in Fig. 7 indicate both the magnitude and direction (positive/negative) of the relationship between HDI and the accuracy of LLMs. The correlation value for DeepSeek-Coder-6.7B-Instruct was found –0.44. This value indicates a negative and moderate correlation between accuracy and HDI. For this model, lower accuracy results were obtained for languages spoken in countries with high HDI values. The correlation value for Qwen2.5Coder-7B-Instruct was found –0.17. This value indicates the presence of both a negative correlation and a very small correlation. In other words, it can be stated that there is a very small relationship between HDI and model accuracy for the Qwen2.5-Coder-7B-Instruct model. For the Lama model, a correlation value of 0.36 was found. This indicates the presence of a weak-medium and positive correlation. This suggests a slight trend towards higher accuracy for languages used in countries with higher HDI values for Llama-3.1-8B-Instruct. Finally, no correlation could be identified for the GPT-5 model due to its 100 % accuracy in all languages, i.e., there was no measurable variance for this model. When all these findings are evaluated, it is found that open-source models exhibit slight fluctuations in accuracy in a positive or negative direction related to HDI, while (GPT-5), a closed model, shows consistency across languages independent of HDI.

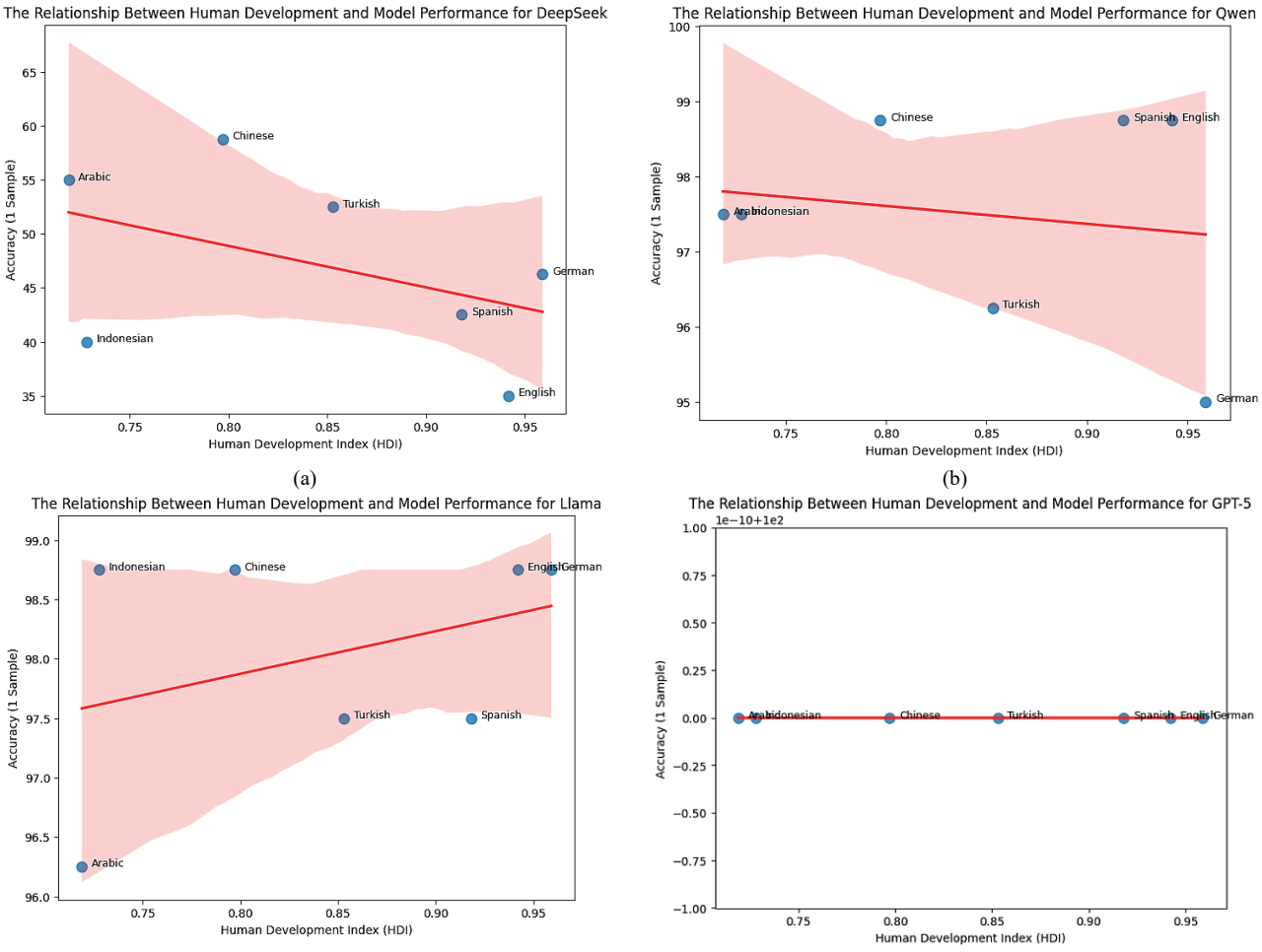


Fig. 7. Correlation between HDI and model performance ($\text{Accuracy}_{(1\text{ sample})}$) across four LLMs: (a) DeepSeek-Coder-6.7B-Instruct (r = –0.44), (b) Qwen2.5-Coder-7B-Instruct (r = –0.17), (c) Llama-3.1-8B-Instruct (r = 0.36), and (d) GPT-5 (r = NaN).

Figure 8 shows the relationship between the number of speakers of languages [58] and the performance of LLMs as a different metric. The number of speakers refers to the total number of individuals who speak the relevant language as a first (L1) or second language (L2) [58]. This value is derived from Ethnologue [59] data on Wikipedia (L1 + L2) [58]. The reason for using this metric additionally is to answer the question: Could this factor have an impact on the success of LLMs, given that languages with large speaker communities are more widespread globally and have a large user base? The visualisation here is based on the prompting technique that gives the highest Accuracy value for each LLM. The bubble size in the figures shows the number of Wikipedia articles in the relevant language. For DeepSeek-Coder-6.7B-Instruct, the graph shows higher accuracy rates for Chinese, which has a higher number of speakers, and Arabic, which has a lower number of speakers, and lower accuracy rates for English, which has a higher number of speakers, and Indonesian, which has a lower number of speakers. The Qwen2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct models showed a weaker dependency between number of speakers and accuracy. In contrast, GPT-5 exhibited accuracy values in all languages, showing a flat accuracy distribution with no noticeable variation, unaffected by the number of speakers. Collectively, the figures show that relatively smaller open-source models demonstrate slight performance differences that may be related to data representation or linguistic prevalence, while larger models such as GPT-5 particularly maintain cross-language consistency regardless of the size of the speaker population or the number of Wikipedia entries.

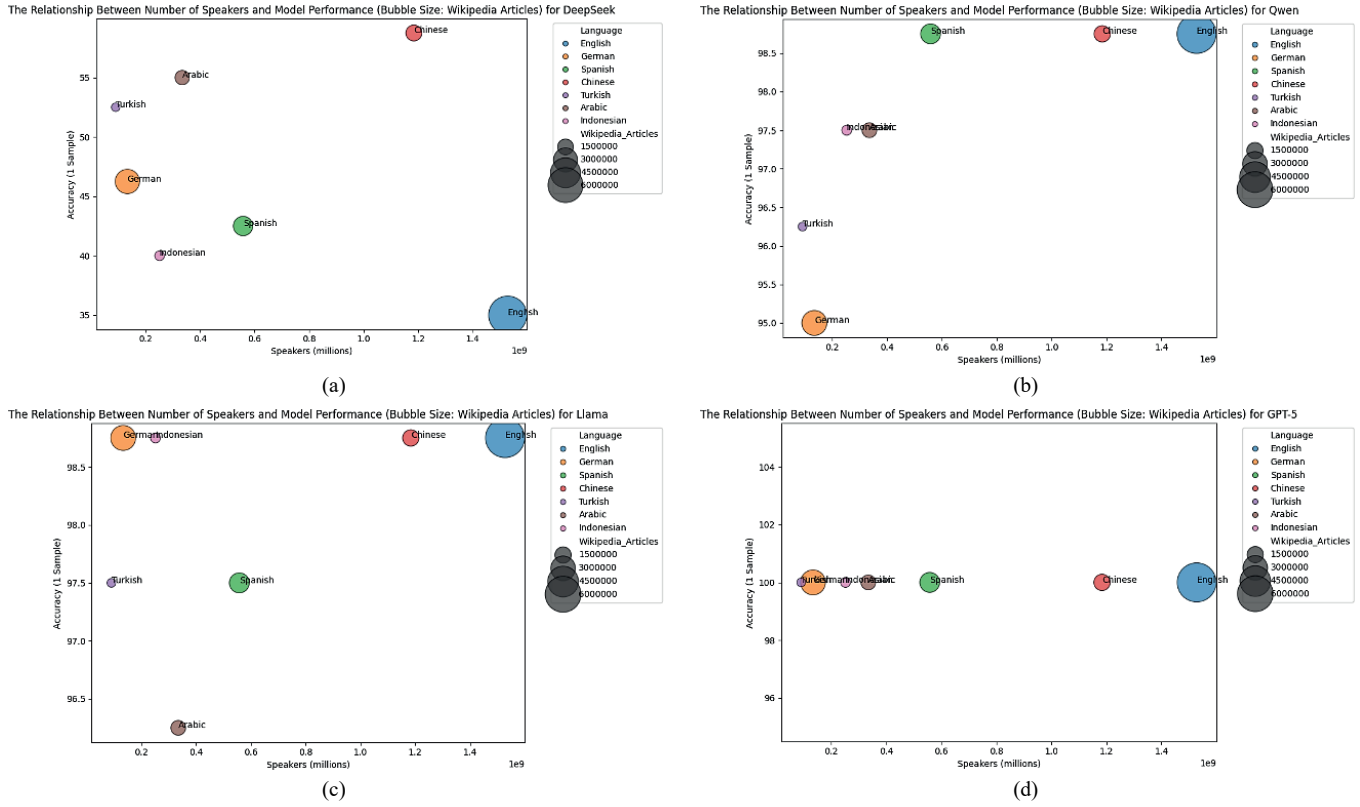


Fig. 8. Relationship between number of speakers and model performance (Accuracy$_{(1 sample)}$) for four LLMs: (a) DeepSeek-Coder-6.7B-Instruct, (b) Qwen2.5-Coder-7B-Instruct, (c) Llama-3.1-8B-Instruct, and (d) GPT-5. Bubble size represents the number of Wikipedia articles per language.

IV. Discussion

A. Effect of Prompt Technique

The performed analyses reveal performance differences between open-source models and GPT-5 depending on the prompting technique used. This difference is especially apparent for a relatively small model such as DeepSeek-Coder-6.7B. While the ZSP technique was the most reliable prompting approach for this model, the CoT technique yielded the best results in most cases for the medium-sized open-source models. The ZSP and FSP techniques yielded more variable and generally lower results. GPT-5, a closed-source, large-scale model, achieved 100 % accuracy in all languages with the CoT technique. Accuracy rates were lower for the ZSP and FSP techniques. These results suggest that the prompting technique can still have a minor impact on accuracy in closed-source high-level models. Therefore, the correct selection of the prompting technique retains its significance in terms of maximising accuracy in GPT-5.

Overall, these findings show that prompting strategies are strongly correlated with model size. While the discrepancies generally narrow as the model capacity increases, the choice of prompting technique can still impact accuracy significantly, even in high-level models. Therefore, the selection of the prompt technique is important in terms of accuracy

optimisation, as well as efficiency, transparency, and interpretability.

B. Cross-Linguistic Performance and Model Generalization

When LLMs are tested across languages, they are found to perform differently (good in some languages, poor in others). The differences in success rates across languages are greater for open-source and smaller-sized LLMs. This suggests that small LLMs are more sensitive to language-specific representational styles. In contrast, medium-sized open-source LLMs consistently achieve high accuracy across different languages. At the same time, the differences between languages in medium-sized LLMs are smaller than in small LLMs. This suggests that greater stability in multilingual generalization is achieved by increasing model size. On the other hand, with closed-source models at the highest level, performance becomes almost language-independent. When considered together, these observations suggest a trend towards scale-based convergence to linguistic neutrality, as the diversity of training data and model capacities increase.

C. Impact of HDI and Linguistic Factors

There is generally a direct correlation between a country's level of development and its access to technology [60], [61]. Languages spoken in countries with high HDI or digitally advanced languages generally have richer and more balanced sources of data. LLMs trained by prioritizing these languages also have the means to learn from diverse and high-quality sources. Conversely, languages spoken in regions with low HDI are underrepresented in large-scale datasets [61], [62]. This situation highlights global inequalities in terms of data access and technological participation.

The findings of the study indicate a correlation, albeit weak, between the performance of open-source LLMs across languages and the HDI values of the countries where the languages are spoken. It was found that the metrics such as the number of speakers of a language as another social indicator, and the number of Wikipedia articles written in the said language as a digital indicator, were not the independent determining factors for the success of LLM's. These two metrics can be jointly influential depending on the structure of the model. The actual decider of the outcome here is the strength of the model and the way in which it is trained. The same amount of data can yield low results for a weak model, high results for a robust model trained in a balanced manner. In other words, if the model capacity is small/medium, as in DeepSeek-Coder-6.7B-Instruct and Qwen2.5-Coder-7B-Instruct, data imbalances within the language will become more noticeable, and the performance will vary. If the model has undergone a large and well-balanced training process, it will 'compensate' for these discrepancies, rendering metrics such as the number of speakers or resource abundance less relevant. Indeed, the GPT-5 model demonstrated performance almost completely independent of socioeconomic development variables. With this model, particularly with the use of a suitable prompting technique, similar accuracy levels were observed regardless of the language's HDI, the number of speakers, or the amount of digital data. These findings suggest that linguistic and developmental asymmetries persist mainly in small and medium-sized models. Therefore, socioeconomic and cultural differences appear to have a particular impact within the domains of small and medium-sized LLMs.

D. Open-Source vs. Closed-Source Models: Robustness and Scaling

Open- and closed-source language models represent two distinct ways of thinking and designing regarding how fault-tolerant the systems are (robustness) and how seamlessly they can be scaled up (scalability). Being transparent and community-supported, open-source LLMs, such as DeepSeek-Coder-6.7B-Instruct, Qwen2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct, represent a rapidly evolving and consistently improving multilingual and task-oriented performance approach. However, due to data scarcity, model size, and resource differences in different languages, accuracy fluctuates slightly across languages, especially for small and medium-sized models.

On the other hand, since they are trained at a very large scale and constantly improved, closed-source models like GPT-5 can produce more accurate results with different prompting techniques and in different languages. However, this consistency comes with less transparency. Information such as the used data, model structure, and details of fine-tuning are kept confidential. This distinction illustrates the relationship between performance and the need for openness/transparency. Open-source models strengthen technological capacity by providing a fair development and use environment where everyone can contribute. However, the lack of finance and deficiencies in infrastructure constrain the progress of these models. Closed-source LLMs, on the other hand, are reliable but reduce transparency, thereby increasing their dependence on companies. As models grow, bridging this gap between open- and closed-source models becomes even more crucial. Therefore, the development of large and accurate models should not be the only goal, if permanent progress is to be achieved. Organisations should deem models/data accessible, measure performance through open-reproducible tests and intentionally add low-resource languages into training and evaluation.

E. Language Prioritization and Developmental Bias

Experimental results reveal that the languages prioritized during the development of LLMs (language focus) coincide with the languages in which they perform best. For example, models like DeepSeek-Coder-6.7B-Instruct and Qwen2.5-Coder-7B-Instruct, trained with a particular emphasis on Chinese and English, DeepSeek-Coder-6.7B-Instruct exhibits higher accuracy in Chinese, and Qwen2.5-Coder-7B-Instruct exhibits higher accuracy in Chinese and English languages, while Meta's Llama-3.1-8B-Instruct yielded more consistent results in English and in the European languages. However, it is also observed that the strength of this relationship diminishes as model capacity and data diversity increase. In large-scale, closed-source systems like GPT-5, cross-lingual consistency has been observed depending on the prompting technique. This

suggests that language priorities become less deciding as development scales increase.

The findings indicate that early-stage open-source LLMs may sometimes reflect the linguistic priorities of their developers. This suggests that in later stages, globally balanced and language-inclusive improvements should be implemented, rather than those specific to one or a few languages.

F. Limitations and Future Directions

The study focused on seven natural languages selected from a multilingual dataset. Therefore, it is important to note that the findings may not fully reflect the linguistic diversity of the real world. Consequently, the boundaries of future research must be expanded. Especially considering the scarcity of studies in low-resource languages [63]. It is essential to broaden these analyses to include lower-resource languages, incorporate interactive prompting processes, and validate the results using multilingual datasets (repeating the study with more comprehensive and diverse datasets will enable a more in-depth analysis and understanding in this field, just as it would in other fields [63]) that feature more challenging problems. The success rates of nearly 100 % achieved by some LLMs indicate benchmark saturation. While this proves that current models have mastered basic code generation, it also indicates that future work should focus on more complex logical reasoning tasks.

Crucially, as these LLMs continue to scale up, securing transparency, inclusiveness, and equitable access becomes a vital mission. Otherwise, preventing linguistic or developmental inequalities in the future generations of AI systems will simply not be possible.

V. Conclusions

LLMs are one of the most important topics in AI research. Especially in recent times, the analysis of LLM performance difference according to used language and selected prompt technique is the popular topic of academic research. This study analyses the performance of three open-source and one closed-source LLMs comparatively under seven natural different languages and three different prompting techniques.

According to the obtained results, among the open-source LLMs, DeepSeek-Coder-6.7B-Instruct showed notable performance differences across languages and achieved lower accuracy than other models. This shows that the model lags other models regarding data imbalance and language representation. In contrast, Qwen2.5-Coder-7B-Instruct and Llama-3.1-8B-Instruct models, which are also open-source, maintained their accuracy levels in all selected languages and command types in the study, demonstrating that some open-source models have reached a successful stage in multilingual generalization.

GPT-5, a closed-source model included in the study for comparison purposes, also showed consistent accuracy across the languages included in the study and achieved the highest accuracy values for both language and prompt-based models. This shows that universal accuracy can be achieved with larger scale LLMs. Furthermore, the findings show that as model size increases, linguistic consistency becomes stronger, and the effect of prompting techniques decreases.

When the results were evaluated from a social perspective, it was observed that there were two different inferences, especially with the increase in model size. The first is that increasing the number of large and diverse models could reduce accuracy differences across languages over time, thus strengthening the possibility of achieving a more balanced, more inclusive level of global performance. Second, since the best performance is mostly seen in closed-source LLMs, new problems in accessibility and transparency may arise. At the core of these problems lies the logic of open- and closed-source. Open-source LLMs offer us a collaborative development opportunity; because their code is publicly available, we can participate in community development and increase the reproducibility of scientific studies. However, closed-source LLMs generally offer higher performance, they also suffer from significant drawbacks, including limited transparency, a single-vendor lock-in, and limited access to their code.

Correlation analyses that include parameters such as HDI, number of speakers, and digital content density reveal that there are still some inequalities, albeit weaker, in small-scale models. However, as the model scale increases, these inequalities gradually decrease or even disappear. Results clearly show how LLMs progress towards language-independent intelligence. Although there is a directly proportional relationship between model accuracy and the size of LLMs, global inequalities in data access and representation remain a specific problem that needs to be solved. Therefore, the main goal is to ensure that these technological advancements are equitably accessible to everyone in the world. Considering all these results, transparency, inclusiveness and fair access should become the new focus of future work in AI development.

References

[1] İ. Kaya, T. H. Gençtürk, and F. K. Gülağız, "A revolutionary acute subdural hematoma detection based on two-tiered artificial intelligence model," *Ulus. Travma Acil Cerrahi Derg.*, vol. 29, pp. 858–871, Aug. 2023. https://doi.org/10.14744/tjtes.2023.76756

[2] T. H. Gençtürk, F. K. Gülağız, and İ. Kaya, "Detection and segmentation of subdural hemorrhage on head CT images," *IEEE Access*, vol. 12, pp. 82235–82246, Jun. 2024. https://doi.org/10.1109/ACCESS.2024.3411932

[3] T. H. Gençtürk, F. K. Gülağız, and İ. Kaya, "Artificial intelligence and computed tomography imaging for midline shift detection," *Eur. Phys. J. Special Topics*, vol. 234, pp. 4539–4566, Oct. 2025. https://doi.org/10.1140/epjs/s11734-025-01779-6

[4] G. Bharathi Mohan, R. Prasanna Kumar, P. Vishal Krishh, A. Keerthinathan, G. Lavanya, M. K. U. Meghana, S. Sulthana, and S. Doss, "An analysis of large language models: Their impact and potential applications," *Knowl. Inf. Syst.*, vol. 66, pp. 5047–5070, Sep. 2024. https://doi.org/10.1007/s10115-024-02120-8

[5] F. K. Gülağız, "Large language models for machine learning design assistance: Prompt-driven algorithm selection and optimization in diverse supervised learning tasks," *Appl. Sci.*, vol. 15, Oct. 2025, Art. no. 10968. https://doi.org/10.3390/app152010968

[6] A. M. Rahmani, A. Hemmati, and S. Abbasi, "The rise of large language models: Evolution, applications, and future directions," *Eng. Rep.*, vol. 7, no. 9, Sep. 2025, Art. no. e70368. https://doi.org/10.1002/eng2.70368

[7] J. Chen, Z. Liu, X. Huang, C. Wu, Q. Liu, G. Jiang, Y. Pu, Y. Lei, X. Chen, X. Wang, K. Zheng, D. Lian, and E. Chen, "When large language models meet personalization: Perspectives of challenges and opportunities," *World Wide Web*, vol. 27, Jun. 2024, Art. no. 42. https://doi.org/10.1007/s11280-024-01276-1

[8] J. Lin *et al*., "How can recommender systems benefit from large language models: A survey," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, Art. no. 28, pp. 1–47, Jan. 2025. https://doi.org/10.1145/3678004
[9] S. Khanna and X. Li, "Invisible languages of the LLM universe," *arXiv preprint*, Art. no. arXiv:2510.11557, Oct. 2025. https://doi.org/10.48550/arXiv.2510.11557
[10] R. Adams *et al*., "Mapping the potentials and limitations of using generative AI technologies to address socio-economic challenges in LMICs," *VeriXiv*, vol. 2, Apr. 2025, Art. no. 57. https://doi.org/10.12688/verixiv.948.1
[11] A. M. Kondoro, "AI writing assistants in Tanzanian universities: Adoption trends, challenges, and opportunities," in *Proc. Fourth Workshop Intell. Interactive Writing Assistants*, Albuquerque, New Mexico, US, May 2025, pp. 37–46. https://doi.org/10.18653/v1/2025.in2writing-1.4
[12] C. Wang *et al*., "Uncovering inequalities in new knowledge learning by large language models across different languages," *arXiv preprint*, Art. no. arXiv:2503.04064, Mar. 2025. https://doi.org/10.48550/arXiv.2503.04064
[13] Z. Li, Y. Shi, Z. Liu, F. Yang, A. Payani, N. Liu, and M. Du, "Language ranker: A metric for quantifying LLM performance across high and low-resource languages," in *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, Philadelphia, Pennsylvania, Apr. 2025, pp. 28186–28194. https://doi.org/10.1609/aaai.v39i27.35038
[14] I. Adebara, H. O. Toyin, N. T. Ghebremichael, A. Elmadany, and M. Abdul-Mageed, "Where are we? Evaluating LLM performance on African languages," *arXiv preprint*, Art. no. arXiv:2502.19582, Feb. 2025. https://doi.org/10.48550/arXiv.2502.19582
[15] I. G. Ilin, "Constitutional-legal aspect of creating large language models: The problem of digital inequality and linguistic discrimination," *J. Digital Technol. Law*, vol. 3, no. 1, pp. 89–107, 2025. https://doi.org/10.21202/jdtl.2025.4
[16] CULTAI Independent Expert Group, "Report of the Independent Expert Group on 2025 Artificial Intelligence and Culture," UNESCO. [Online]. Available: https://www.unesco.org/sites/default/files/medias/fichiers/2025/09/CULTAI_Report%20of%20the%20Independent%20Expert%20Group%20on%20Artificial%20Intelligence%20and%20Culture%20%28final%20online%20version%29%201.pdf [Accessed Oct. 25, 2025].
[17] C. Zhang, M. Tao, Z. Liao, and Y. Feng, "MiLiC-Eval: Benchmarking multilingual LLMs for China's minority languages," *arXiv preprint*, Art. no. arXiv:2503.01150, Jun. 2025. https://doi.org/10.48550/arXiv.2503.01150
[18] I. A. Azime *et al*., "Proverbeval: Exploring LLM evaluation challenges for low-resource language understanding," *arXiv preprint*, Art. no. arXiv:2411.05049, Feb. 2025. https://doi.org/10.48550/arXiv.2411.05049
[19] O. Khade, S. Jagdale, A. Phaltankar, G. Takalikar, and R. Joshi, "Challenges in adapting multilingual LLMs to low-resource languages using LoRA PEFT tuning," *arXiv preprint*, Art. no. arXiv:2411.18571, Nov. 2024. https://doi.org/10.48550/arXiv.2411.18571
[20] D. Bordonaba-Plou and L. M. Jreis-Navarro, "Linguistic injustice in multilingual technologies: The TenTen Corpus Family as a case study," in *Multilingual Digital Humanities*, 1st ed. Routledge, 2023, pp. 129–144. https://doi.org/10.4324/9781003393696-12
[21] S. Atreja, J. Ashkinaze, L. Li, J. Mendelsohn, and L. Hemphill, "What's in a prompt? A large-scale experiment to assess the impact of prompt design on the compliance and accuracy of LLM-generated text annotations," in *Proc. Int. AAAI Conference on Web and Social Media*, Copenhagen, Denmark, Jun. 2025, pp. 122–145. https://doi.org/10.1609/icwsm.v19i1.35807
[22] E. Chen, "Enhancing teaching quality through LLM: An experimental study on prompt engineering," in *Proc. 14th Int. Conf. Educ. Inf. Technol. (ICEIT)*, Guangzhou, China, May 2025, pp. 1–7. https://doi.org/10.1109/ICEIT64364.2025.10976127
[23] R. Khojah, F. G. de Oliveira Neto, M. Mohamad, and P. Leitner, "The impact of prompt programming on function-level code generation," *IEEE Trans. Softw. Eng.*, vol. 51, no. 8, pp. 2381–2395, Aug. 2025. https://doi.org/10.1109/TSE.2025.3587794
[24] Q. Ma, W. Peng, C. Yang, H. Shen, K. Koedinger, and T. Wu, "What should we engineer in prompts? Training humans in requirement-driven LLM use," *ACM Trans. Comput.-Hum. Interact.*, vol. 32, no. 4, pp. 1–27, Aug. 2025. https://doi.org/10.1145/3731756
[25] T. Debnath, M. N. A. Siddiky, M. E. Rahman, P. Das, A. K. Guha, M. R. Rahman, and H. M. D. Kabir, "A comprehensive survey of prompt engineering techniques in large language models," *TechRxiv*, Oct. 2025. https://doi.org/10.36227/techrxiv.174140719.96375390/v1
[26] L. J. Jacobsen and K. E. Weber, "The promises and pitfalls of large language models as feedback providers: A study of prompt engineering and the quality of AI-driven feedback," *AI*, vol. 6, no. 2, Feb. 2025, Art. no. 35. https://doi.org/10.3390/ai6020035
[27] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, "Unleashing the potential of prompt engineering for large language models," *Patterns*, vol. 6, no. 6, Jun. 2025, Art. no. 101260. https://doi.org/10.1016/j.patter.2025.101260
[28] T. S. Almeida, G. K. Bonás, J. G. A. Santos, H. Abonizio, and R. Nogueira, "TiEBe: Tracking language model recall of notable worldwide events through time," *arXiv preprint*, Art. no. arXiv:2501.07482, May 2025. https://doi.org/10.48550/arXiv.2501.07482
[29] Q. Peng, Y. Chai, and X. Li, "HumanEval-XL: A multilingual code generation benchmark for cross-lingual natural language generalization," in *Proc. 2024 Joint Int. Conf. Comput. Linguist., Lang. Resour. Eval. (LREC-COLING 2024)*, Torino, Italia, May 2024, pp. 8383–8394. [Online]. Available: https://aclanthology.org/2024.lrec-main.735/
[30] M. Chen *et al*., "Evaluating large language models trained on code," *arXiv preprint*, Art. no. arXiv:2107.03374, Jul. 2021. https://doi.org/10.48550/arXiv.2107.03374
[31] Wikimedia Foundation, "List of Wikipedias," *Meta-Wiki*. [Online]. Available: https://meta.wikimedia.org/wiki/List_of_Wikipedias [Accessed Oct. 25, 2025].
[32] United Nations Development Programme, "Human development report 2025: A matter of choice: People and possibilities in the age of AI," United Nations Development Programme, New York. [Online]. Available: https://hdr.undp.org/content/human-development-report-2025 [Accessed Oct. 25, 2025].
[33] Hugging Face, "deepseek-coder-6.7b-instruct," *Hugging Face*. [Online]. Available: https://huggingface.co/deepseek-ai/deepseek-coder-6.7b-instruct [Accessed Oct. 25, 2025].
[34] B. Hui *et al*., "Qwen2.5-Coder technical report," *arXiv preprint*, Art. no. arXiv:2409.12186, Sep. 2024. https://doi.org/10.48550/arXiv.2409.12186
[35] Hugging Face, "Llama-3.1-8B-Instruct," *Hugging Face*. [Online]. Available: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct [Accessed Oct. 25, 2025].
[36] OpenAI, "GPT-5 system card," *OpenAI*. [Online]. Available: https://cdn.openai.com/gpt-5-system-card.pdf [Accessed Oct. 25, 2025].
[37] DeepSeek, "Quick start – The temperature parameter," *DeepSeek API Docs*. [Online]. Available: https://api-docs.deepseek.com/quick_start/parameter_settings [Accessed Oct. 25, 2025].
[38] C. Arora, A. I. Sayeed, S. Licorish, F. Wang, and C. Treude, "Optimizing large language model hyperparameters for code generation," *arXiv preprint*, Art. no. arXiv:2408.10577, Aug. 2024. https://doi.org/10.48550/arXiv.2408.10577
[39] S. Ouyang, J. M. Zhang, M. Harman, and M. Wang, "An empirical study of the non-determinism of ChatGPT in code generation," *arXiv preprint*, Art. no. arXiv:2308.02828, Oct. 2024. https://doi.org/10.48550/arXiv.2308.02828
[40] DeepSeek, "DeepSeek Coder," *GitHub*. [Online]. Available: https://github.com/deepseek-ai/DeepSeek-Coder [Accessed Oct. 25, 2025].
[41] DeepSeek, "DeepSeek Coder: Let the code write itself," DeepSeek Coder Website. [Online]. Available: https://deepseekcoder.github.io/ [Accessed Oct. 25, 2025].
[42] D. Guo *et al*., "DeepSeek-Coder: When the large language model meets programming – The rise of code intelligence," *arXiv preprint*, Art. no. arXiv:2401.14196, Jan. 2024. https://doi.org/10.48550/arXiv.2401.14196
[43] A. Yang *et al*., "Qwen 2.5 technical report," *arXiv preprint*, Art. no. arXiv:2412.15115, Dec. 2024. https://doi.org/10.48550/arXiv.2412.15115
[44] Hugging Face, "Llama 3.1 – 405B, 70B & 8B with multilinguality and long context," *Hugging Face Blog*. [Online]. Available: https://huggingface.co/blog/llama31 [Accessed Oct. 25, 2025].
[45] Meta, "meta-llama/llama3," *GitHub*. [Online]. Available: https://github.com/meta-llama/llama3 [Accessed Oct. 25, 2025].
[46] OpenAI, "Introducing GPT-5," *OpenAI*. [Online]. Available: https://openai.com/tr-TR/index/introducing-gpt-5/ [Accessed Aug. 7, 2025].

[47] T. Brown *et al*., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst. 33 (NeurIPS 2020)*, Virtual-only conference, Art. no. 159, Dec. 2020, pp. 1877–1901. [Online]. Available: https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html
[48] J. Wei *et al*., "Chain-of-thought prompting elicits reasoning in large language models," in *Adv. Neural Inf. Process. Syst.*, Art. no. 1800, Nov. 2022, pp. 24824–24837. [Online]. Available: https://dl.acm.org/doi/10.5555/3600270.3602070
[49] Q. Dong *et al*., "A survey on in-context learning," *arXiv preprint*, Art. no. arXiv:2301.00234, Oct. 2024. https://doi.org/10.48550/arXiv.2301.00234
[50] Y. Li, "A practical survey on zero-shot prompt design for in-context learning," in *Proc. 14th Int. Conf. Recent Advances in Natural Language Processing* (RANLP), Varna, Bulgaria, Sep. 2023, pp. 641–647. https://doi.org/10.26615/978-954-452-092-2_069
[51] J. Austin *et al*., "Program synthesis with large language models," *arXiv preprint*, Art. no. arXiv:2108.07732, Aug. 2021. https://doi.org/10.48550/arXiv.2108.07732
[52] A. Lewkowycz *et al*., "Solving quantitative reasoning problems with language models," in *Adv. Neural Inf. Process. Syst.*, Art. no. 278, Nov. 2022, pp. 3843–3857. [Online]. Available: https://dl.acm.org/doi/10.5555/3600270.3600548
[53] H. Zhou, A. Nova, H. Larochelle, A. Courville, B. Neyshabur, and H. Sedghi, "Teaching algorithmic reasoning via in-context learning," *arXiv preprint*, Art. no. arXiv:2211.09066, Nov. 2022. https://doi.org/10.48550/arXiv.2211.09066
[54] H. Xia, C. T. Leong, W. Wang, Y. Li, and W. Li, "Tokenskip: Controllable chain-of-thought compression in LLMs," *arXiv preprint*, Art. no. arXiv:2502.12067, Sep. 2025. https://doi.org/10.48550/arXiv.2502.12067
[55] M. Franke, "Sheet 6.3: Decoding strategies," *Pragmatic Natural Language Generation with Neural Language Models Web Site*. [Online]. Available: https://michael-franke.github.io/npNLG/06-LSTMs/06d-decoding-GPT2.html [Accessed Oct. 25, 2025].
[56] Hugging Face, "Generation strategies – Transformers documentation," *Hugging Face*. [Online]. Available: https://huggingface.co/docs/transformers/en/generation_strategies [Accessed Oct. 25, 2025].
[57] Q. Luo, F. Hariri, L. Eloussi, and D. Marinov, "An empirical analysis of flaky tests," in *Proc. 22nd ACM SIGSOFT Int. Symp. Found. Software Eng.*, Hong Kong, China, Nov. 2014, pp. 643–653. https://doi.org/10.1145/2635868.2635920
[58] Wikipedia, "List of languages by total number of speakers," *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers [Accessed Oct. 25, 2025].
[59] D. M. Eberhard, G. F. Simons and C. D. Fennig, Eds., *Ethnologue: Languages of the World*, 28th ed. SIL International, 2025. [Online]. Available: https://www.ethnologue.com/insights/ethnologue200/ [Accessed Oct. 25, 2025].
[60] P. Bala, "The impact of mobile broadband and internet bandwidth on human development – A comparative analysis of developing and developed countries," *J. Knowl. Econ.*, vol. 15, pp. 16419–16453, Jan. 2024. https://doi.org/10.1007/s13132-023-01711-0
[61] World Bank, "Chapter 1 – Digital adoption: Accelerating post-pandemic, yet a widening divide," in *Digital Progress and Trends Report 2023*, World Bank. [Online]. Available: https://www.worldbank.org/en/publication/digital-progress-and-trends-report?utm_source=https://www.google.com/search?#Report_chapters [Accessed Oct. 25, 2025].
[62] World Bank, "Chapter 5 – Artificial intelligence: Revolutionary potential and huge uncertainties," in *Digital Progress and Trends Report 2023*, World Bank. [Online]. Available: https://www.worldbank.org/en/publication/digital-progress-and-trends-report?utm_source=https://www.google.com/search?#Report_chapters [Accessed Oct. 25, 2025].
[63] A. Daupare and G. Jēkabsons, "Benchmarking 24 large language models for automated multiple-choice question generation in Latvian," *Applied Computer Systems*, vol. 30, no. 1, pp. 85–90, May 2025. https://doi.org/10.2478/acss-2025-0010

Fidan Kaya Gülağız received the B.Sc., M.S., and PhD degrees in Computer Engineering from Kocaeli University, Turkey, in 2010, 2012, and 2018, respectively. Her doctoral research focused on estimation of synchronization time in content delivery networks (CDNs) using the Profile Hidden Markov Model (PHMM). She is currently an Assistant Professor with the Department of Computer Engineering, Kocaeli University. Her main research interests include artificial intelligence, machine learning, and distributed systems.
E-mail: fidan.kaya@kocaeli.edu.tr / fdnkaya@gmail.com
ORCID iD: https://orcid.org/0000-0003-3519-9278