UDC 004.8:004.4

MULTILINGUAL REPOSITORY-LEVEL CODE QUESTION ANSWERING BENCHMARK

Syromiatnikov M. V.
PhD, Professor of the Software Engineering Department Ruvinska V. M.
Odesa Polytechnic National University, UKRAINE

ABSTRACT. This paper presents a multilingual benchmark for repository-level code question answering. The benchmark includes English, German, and Ukrainian queries derived from open-source repositories, paired with English reference answers and file-level evidence. A full-context large language model baseline is evaluated as an upper bound, where the model processes the entire repository context. The results are analyzed by language, repository size, context level, and evidence quality to assess the complexity of multilingual repository understanding tasks.

Introduction. Large language models (LLMs) are now used in software engineering for code generation, review, documentation support, and maintenance tasks [1]. Many of these scenarios cannot be solved with a short code fragment, as they require configs, data models, and component interactions to be considered together. This makes repository-level context necessary, but difficult to use, especially when repositories contain many files and relevant evidence is scattered. Multilingual question answering (QA) adds another difficulty: repositories are usually written mostly in English, while users may ask questions in other languages. Recent benchmarks address repository-level QA [2], issue resolution [3], and code review evaluation [4]. However, the combination of multilingual query variants, QA-oriented context levels, and maintenance-oriented reasoning remains insufficiently covered. Therefore, this work introduces a multilingual, evidence-grounded benchmark for repository-level code QA.

Objective. The aim of this research is to develop a multilingual, evidence-based benchmark for repository-level code question answering to evaluate how LLMs support software maintenance tasks that require different levels of repository context. The benchmark is intended for tasks where the answer cannot be derived from a short code fragment and requires evidence from files, components, configuration, tests, or implementation flows. To achieve this aim, the following tasks are defined:

- to select open-source repositories suitable for repository-level question answering evaluation under a fixed context-size constraint;
- to form a set of multilingual question-answer pairs in English, German, and Ukrainian with a context level, topic, answer type, difficulty label, and file-level evidence specified for each pair;
- to evaluate a full-context large language model baseline as an upper-bound setting;
- to examine the results by language, repository size, context level, and evidence quality.

The main part of the work. The benchmark was constructed in three stages: repository selection, multilingual question-answer generation, and full-context baseline evaluation. The repository set includes 30 open-source projects written mainly in Python, Go, JavaScript/TypeScript, PHP, Java, and Rust. The selected projects include web applications, services, demonstration systems, frameworks, and supporting libraries. Very large repositories and generated or dependency-heavy folders were excluded because the experiment required each repository to fit into the LLM context window.

The generation procedure follows the idea of synthetic question-answer annotation used in earlier context-based question-answering work, where LLMs were applied to generate question-answer pairs from a given context when manual annotation was limited [5]. In this study, this procedure was adapted from text paragraphs to software repositories. For each repository, a set of question trajectories and topic targets was prepared to reflect realistic software flows, such as routing, configuration, persistence, validation, testing, and maintenance impact. These trajectories, together with the complete filtered repository context, were provided to Claude Opus 4.7, which generated 30 canonical QA items per repository. Each item contained English, German, and Ukrainian question variants, one English reference answer, and file-level evidence. The same evidence is used for all language variants, so the language changes while the underlying repository fact remains fixed.

The questions are divided into four context levels. L1 local questions are answerable from one file. L2 cross-file questions require several related files. L3 repository-flow questions require understanding

a feature across modules or layers. L4 maintenance questions ask about change impact, quality, tests, or files that should be inspected before modifying behavior. This division is important because repository-level question answering is not a single task: a local configuration question and a maintenance-impact question require different amounts of context. A simplified example of a benchmark item is shown in Figure 1.


```
{
  "question_id": "repo_0028",
  "question_intent": "Ask how token usage is tracked across pipeline stages and persisted.",
  "question_en": "How is LLM token usage tracked across pipeline stages and where is it
     persisted?",
  "question_de": "Wie wird die LLM-Token-Nutzung über Pipeline-Stufen hinweg verfolgt und wo
     wird sie gespeichert?",
  "question_uk": "Як відстежується використання токенів LLM на різних етапах пайплайна та де
     воно зберігається?",
  "reference_answer_en": "main._TokenUsageTracker accumulates usage per stage (activation,
     scoring, baseline, etc.) and a global total, recording each call's provider, model, phase
     , kind, and cache_hit. run_pipeline (and baseline_full_repo.run_baseline) instantiate it
     with a token_usage.json path inside the work directory, call tracker.add(stage=..., usage
     =...) after every LLMRuntime.call_json_cached_with_usage, and call tracker.flush() to
     write the file. The final report includes token_usage_path and token_usage_total fields
     summarizing the totals.",
  "evidence": [
    {"path": "main.py", "role": "Defines _TokenUsageTracker and integrates it with run_pipeline
       LLM calls."},
    {"path": "llm_runtime.py", "role": "Returns usage dictionaries from
       call_json_cached_with_usage that the tracker consumes."},
    {"path": "baseline_full_repo.py", "role": "Reuses _TokenUsageTracker for the single
       baseline call and writes token_usage.json."}
  ],
  "context_level": "L3_repository_flow",
  "topic": "data_model_or_persistence",
  "answer_type": "step_by_step_flow",
  "difficulty": "medium",
  "expected_file_count": 3
}
```


Figure 1 – Sample structure of a benchmark item

The generated dataset initially contained 2700 language-specific QA pairs. Manual review identified 198 language-specific variants with minor or major issues related to question formulation, context consistency, reference answers, or evidence files. These variants belonged to 89 canonical multilingual QA items. To keep the multilingual comparison fair, the whole canonical item was excluded if any of its language variants contained an issue. Therefore, 267 language-specific pairs were excluded in total, and the final evaluated dataset contained 2433 language-specific QA pairs.

The full-context baseline was run with GPT-5.4 using medium reasoning effort. For each repository and language, the model received the complete filtered repository context and all questions in that language. The answers were then evaluated with a separate GPT-5.5 judge following the LLM-as-a-Judge setup [6], and the resulting scores were manually checked. For each item, the judge received the question, reference answer, generated answer, gold evidence, generated evidence paths, and the contents of files referenced by either side. Two scores were assigned: answer correctness and evidence quality. The answer score reflects semantic correctness and completeness of the generated answer, while the evidence score measures how well the returned file paths cover the files needed to support the answer.

The judge first evaluated the answers on a 0-4 scale. For answer correctness: 4 denotes a correct and sufficiently complete answer, 3 a mostly correct answer with minor omissions or inaccuracy, 2 a partially correct answer with important missing information or mixed statements, 1 an answer with only small relevant fragments, and 0 a completely incorrect or hallucinated answer. For evidence quality: 4 means that the returned file paths cover all or almost all necessary evidence, 3 means mostly complete evidence with minor omissions, 2 means partial evidence, 1 means weakly relevant evidence, and 0 means missing or hallucinated evidence. The integer scores were then normalized to the 0-1 range for reporting in Table 1.

Table 1 – Full-context baseline results by repository size and context level

Repository size	Context level	English		German		Ukrainian		Questions per language
		Answer	Evidence	Answer	Evidence	Answer	Evidence	
Small	L1 local	**0.990**	1.0	0.986	1.0	0.979	1.0	72
	L2 cross-file	**0.968**	**0.940**	**0.968**	0.932	0.965	0.917	100
	L3 repository-flow	**0.959**	**0.907**	0.924	0.884	0.948	0.872	43
	L4 maintenance	**0.859**	**0.848**	**0.859**	0.826	0.837	**0.848**	23
Medium	L1 local	0.982	1.0	**0.986**	1.0	0.975	1.0	70
	L2 cross-file	**0.962**	**0.936**	0.955	0.927	0.942	0.910	117
	L3 repository-flow	0.950	**0.869**	**0.965**	0.835	0.938	0.842	65
	L4 maintenance	0.812	0.812	**0.863**	**0.850**	0.850	0.838	20
Large	L1 local	0.977	1.0	0.968	1.0	**0.980**	1.0	87
	L2 cross-file	**0.961**	**0.870**	**0.961**	0.861	0.940	0.845	108
	L3 repository-flow	**0.937**	0.823	0.927	**0.839**	0.921	0.807	79
	L4 maintenance	0.806	0.759	0.815	**0.769**	**0.833**	**0.769**	27

Conclusions. This work introduced a multilingual benchmark for repository-level code question answering over 30 open-source repositories. The benchmark contains English, German, and Ukrainian questions, English reference answers, and file-level evidence. The questions cover four context levels: local, cross-file, repository-flow, and maintenance-impact. The final dataset is publicly available at https://huggingface.co/datasets/NLPForUA/multilingual-repo-qa.

The full-context GPT-5.4 baseline achieved high scores in all three languages. The weighted normalized answer score was 0.953 for English, 0.951 for German, and 0.944 for Ukrainian. Local questions were almost fully solved, with an average answer score close to 0.98 and an evidence score equal to 1.00. However, the results decreased for more complex questions. For L4 maintenance questions, answer scores ranged from 0.806 to 0.863, and evidence scores ranged from 0.759 to 0.850. This indicates that the main difficulty is not only producing a correct answer, but also grounding it in the complete set of relevant files.

Finally, the full-context setting should be treated as a strong upper-bound baseline rather than a practical deployment scenario, as the model has access to the entire repository rather than only retrieved or partial context. Future work will compare retrieval-based, schema-based, and fine-tuned approaches with this baseline as more practical approximations of repository-level QA performance.

REFERENCES

1. Zhang Q., Fang C., Xie Y. et al. A survey on large language models for software engineering. *Science China Information Sciences*. 2026. Vol. 69. Article 141102. DOI: 10.1007/s11432-025-4670-0.
2. Alebachew Y. B., Leary H., Vaishampayan S., Brown C. Beyond Code Snippets: Benchmarking LLMs on Repository-Level Question Answering. *Companion Proceedings of the 34th ACM Symposium on the Foundations of Software Engineering (FSE Companion 2026)*. 2026. DOI: 10.1145/3803846.3807467.
3. Wang L., Ramalho L., Celestino A. et al. SWE-Bench++: A Framework for the Scalable Generation of Software Engineering Benchmarks from Open-Source Repositories. *arXiv preprint*. 2025. DOI: 10.48550/arXiv.2512.17419.
4. Khan T. I., Wang S., Zhang H., Chen T.-H. A Survey of Code Review Benchmarks and Evaluation Practices in Pre-LLM and LLM Era. *arXiv preprint*. 2026. DOI: 10.48550/arXiv.2602.13377.
5. Syromiatnikov M. V., Ruvinskaya V. M. UA-LLM: Advancing Context-Based Question Answering in Ukrainian through Large Language Models. *Radio Electronics, Computer Science, Control*. 2024. No. 1. P. 147–160. DOI: 10.15588/1607-3274-2024-1-14.
6. Gu J., Jiang X., Shi Z. et al. A Survey on LLM-as-a-Judge. *arXiv preprint*. 2024. DOI: 10.48550/arXiv.2411.15594.