DOI: https://doi.org/10.15276/ict.03.2026.18
UDC 004.8:004.4

Evaluating the reliability of function calling by large language models for ukrainian-language requests

Oleksii I. Sheremet[1)]
ORCID: https://orcid.org/0000-0003-1298-3617; oleksii.sheremet@ddma.edu.ua. Scopus Author ID: 57170410800
Roman S. Tomashevskyi[1)]
ORCID: https://orcid.org/0000-0002-5278-9272; roman.tomashevskyi@ddma.edu.ua. Scopus Author ID: 56338488300
Oksana V. Chmykhova[1)]
ORCID: https://orcid.org/0000-0002-9198-9701; oksana.chmykhova@ddma.edu.ua. Scopus Author ID: 57194619495
Bohdan V. Vorobiov[1)]
ORCID: https://orcid.org/0000-0002-0264-354X; bohdan.vorobiov@ddma.edu.ua. Scopus Author ID: 57891861500
[1)] Donbas State Engineering Academy, 72, Akademichna St. Kramatorsk, 84313, Ukraine

ABSTRACT

Function calling by large language models is increasingly used to connect natural-language requests to software functions, yet its reliability for Ukrainian-language requests remains poorly characterized. We introduce UA-ToolCall, a paired English–Ukrainian benchmark with twenty bilingual mock-tool contracts, twenty-four development cases, and one hundred and twenty fixed test cases (one hundred required calls, ten clarification cases, and ten no-tool cases). It varies query and description languages, catalogue size, and distractor similarity while keeping machine identifiers canonical. Models from the Qwen3, Granite 3.3, and SmolLM3 families were evaluated in quantized form using a shared text prompt for unconstrained one-step generation of structured function-call outputs and greedy decoding, producing three thousand six hundred outputs. Under English descriptions and the full catalogue of twenty tools, Ukrainian queries reduced strict success from ninety-four to seventy-four percent, from forty-three to nine percent, and from twenty-six to nine percent, respectively; all paired effects remained significant after Holm correction. Qwen kept one hundred percent tool-selection accuracy, so its loss came from arguments and canonicalization; Granite lost selection and arguments; SmolLM3 was dominated by decision-field errors. Ukrainian and bilingual descriptions improved SmolLM3 by eleven and seventeen percentage points, respectively, but did not produce statistically significant gains for the other models; bilingual descriptions increased mean prompt length by fifty-four to eighty-three percent. Within the single catalogue order generated with seed 1701, neither catalogue contrast was statistically significant; order variability was not estimated. These results support reporting semantic and executable correctness separately and indicate that localization benefits are model-dependent. The findings apply to prompt-based one-step generation of structured function-call outputs by the recorded Q4_K_M builds, not native function-calling interfaces, constrained decoding, or multi-step tool interaction.

Keywords: Large language models; function calling; tool use; multilingual evaluation; structured outputs

For citation: **Sheremet O. I., Tomashevskyi R. S., Chmykhova O. V., Vorobiov B. V. "Evaluating the reliability of function calling by large language models for ukrainian-language requests". *Informatics. Culture. Technology*. 2026; Vol. 3 No. 1 (3): 210–220. DOI: https://doi.org/10.15276/ict.03.2026.18**

Introduction. Tool calling turns a language model into a software component that selects operations and populates typed arguments. Its output must satisfy both linguistic intent and a machine contract: function name, JSON structure, field set, data type, identifier, date, currency code, and enum token. A response can therefore be understandable to a person but unusable by the target program. This distinction is particularly relevant to function calling with Ukrainian-language requests. A user may write «із Києва», «42.5», «гривні», or «на завтра», while the contract requires Kyiv, 42.5, UAH, and an ISO date. Such outputs may preserve meaning while violating an operational convention, so evaluation must separate semantic correctness from executable correctness.

Function-calling benchmarks now cover argument accuracy, relevance, parallel calls, and stateful interaction [1], [2], [3]. MASSIVE-Agents evaluates 52 languages and description localization [4], while MLCL separates semantic from execution-level failures in Chinese, Hindi, and Igbo [5]. Neither includes Ukrainian or jointly controls description language and a semantically confusing catalogue. Ukrainian resources cover alignment, general-language, and code-generation evaluation [6], [7], [8], yet as of August 2026 we have found no peer-reviewed benchmark of structured tool invocation in Ukrainian. UA-ToolCall fills that gap by holding meaning constant while varying request language, description language, catalogue size, and distractor similarity.

The objective of this study is to quantify the strict and semantic reliability of one-step prompt-based large language model (LLM) function-call generation for Ukrainian-language requests relative to paired English requests, and to test whether description localization and catalogue composition modify performance. The study asks whether Ukrainian requests reduce strict success (RQ1), whether Ukrainian descriptions improve Ukrainian-query performance (RQ2), how the prespecified catalogue contrasts behave under the single seed-1701 tool order (RQ3), whether bilingual descriptions mitigate failures (RQ4), and which error classes explain the semantic-operational gap (RQ5). One directional hypothesis per question, H1-H5, was fixed before final inference and documented in the companion experimental protocol; the conclusions state which of them the data support.

UA-ToolCall contributes a fully specified paired dataset with canonical values, no-tool decisions, and clarification cases; controlled catalogue confusability; separate strict, semantic, schema, selection, and relevance outcomes; and a documented workflow retaining model revisions, raw outputs, statistics, and figures.

The scope is one-step JSON-object generation under a unified text prompt. The study does not test native vendor function-calling APIs, constrained decoding, multi-step agents, or real tool execution; conclusions are therefore limited to the recorded prompt, model builds, quantization, and catalogue order.

Related work. BFCL established execution- and AST-based evaluation and later added multi-turn behaviour [1]. RoTBench separates tool selection, parameter identification, and content filling under noise [9]; StableToolBench targets unstable external tools [10]; and ToolSandbox tests stateful simulated interactions [3]. HammerBench and ACEBench extend fine-grained tool-use evaluation to mobile-device and broader usage settings [11], [12]. APIGen and ToolACE demonstrate verifiable collection generation [13], [14]. These works motivate deterministic mock contracts and preserved raw outputs.

MASSIVE-Agents provides 47,020 examples in 52 languages, 55 functions, and controlled translation settings [4]. MLCL reports a multilingual execution gap, including correct tool selection with non-canonical localized values [5]. MAPS broadens multilingual agent evaluation to performance and security, although it is not function-calling-specific [15]. Related work covers Traditional Chinese [16], Arabic [17], and Bulgarian adaptation [18]. Ukrainian is absent from these tool-calling studies, so the contribution here is the controlled interaction of language, descriptions, and catalogue conditions.

Tool preference can shift with minor wording changes [19], and semantically similar tools can degrade selection and argument assignment [20]. When2Call adds the decision not to invoke a tool [2]. Structured-output studies also show that schema validity depends on whether it is produced by the model or enforced by a decoder [21]. For that reason UA-ToolCall uses a single unconstrained, protocol-neutral output format. Table 1 positions the benchmark against its closest neighbours. A broader survey of tool learning with foundation models provides the conceptual context for these evaluation concerns [22].

Table 1. **Positioning of UA-ToolCall relative to close benchmarks**

Benchmark	Language and localization	Catalogue and evaluation
BFCL [1]	Mainly English; no localization factor	Multiple task types; AST/execution
MASSIVE-Agents [4]	52 languages; translated instructions/descriptions	Fixed benchmark sets; AST accuracy
MLCL [5]	EN, ZH, HI, IG; partial/full translation	Controlled tools; semantic/executable split
Robustness study [20]	English; query paraphrasing	Similar distractors; selection/arguments
UA-ToolCall	EN/UK; EN, UK, or bilingual descriptions	easy5/hard5/full20; prespecified dual score

Source: **compiled by the authors**

Benchmark design and dataset. The catalogue contains 20 mock functions in ten close pairs: current weather/forecast, exchange rate/conversion, event/reminder, train/bus, hotel search/booking, pharmacy/clinic, parcel price/tracking, order status/cancellation, restaurant search/reservation, and translation/language detection. Each function has bilingual descriptions and a JSON Schema-like contract with required fields, types, patterns, enums, bounds, and additionalProperties=false.

Function and field names remain English machine identifiers. Currency and language codes, IDs, status/reason codes, units, dates, and times remain canonical in every condition; only natural-language descriptions and parameter explanations are localized.

The dataset contains 144 paired intents: 24 development cases and a fixed test split of 100 positive cases (five per function), 10 clarification cases, and 10 no-tool cases. Each item stores paired requests, gold decision and arguments, easy5 and hard5 pools, difficulty, and linguistic tags. Every prompt carries the same synthetic reference timestamp, 20 August 2026, 09:00 Europe/Kyiv (+03:00), so that relative expressions resolve deterministically; this clock is a fixed parameter of the prompt and not the date on which the runs were executed. Test materials were frozen before inference. After inference, all 144 English–Ukrainian query pairs were audited side by side, without reference to the model outputs, using a checklist covering intent, polarity, argument presence, expected decision and tool, dates, canonical values, and prespecified aliases. No discrepancies requiring correction of the data or gold labels were identified. Table 2 summarizes the resulting split. The frozen task records and the dataset manifest underlying this composition are publicly available in the UA-ToolCall repository (data/tasks.jsonl and data/dataset_manifest.json).

Table 2. **Fixed test dataset composition**

Split / decision	Cases	EN–UK pairs	Evaluation
Development / mixed	24	24	Pilot only
Test / tool_call	100	100	Primary and secondary metrics
Test / clarify	10	10	Decision and missing fields
Test / no_tool	10	10	Relevance and unnecessary calls
Total	144	144	120 test cases × 10 conditions per model

Source: **compiled by the authors**

Scenarios cover inflected toponyms, decimal commas, currencies, dates, time zones, optional fields, canonical enums, negation, identifiers, and free text. Gold arguments were revalidated against the schemas during dataset construction; the current build has zero schema errors. Locale aliases such as Київ/Kyiv were fixed before inference.

The catalogue conditions comprise two prespecified five-tool pools and the full 20-tool catalogue. Both five-tool pools are subsets of full20. easy5 contains the intended tool plus four unrelated distractors; hard5 contains the intended tool, its closest competitor, and three action-oriented distractors; full20 contains every tool. A SHA-256 rank derived from seed 1701 fixes global order across paired conditions. Runs were grouped by catalogue prefix only to reuse the llama.cpp key-value cache; prompts and scoring were unchanged. This design contrasts two distractor sets at fixed size and compares hard5 with the larger full20 catalogue. Because only this order was run, all RQ3 contrasts are specific to the seed-1701 configuration and do not estimate order variability.

Experimental setup. The 2 × 2 × 2 core design crosses query language (EN/UK), description language (EN/UK), and catalogue (hard5/full20). Two additional prespecified conditions (UK query/EN descriptions/easy5 and UK query/bilingual descriptions/full20) give 10 conditions, 120 × 10 = 1,200 responses per model and 3,600 overall.

All three models used Q4_K_M GGUF builds. The complete reproducibility package – including the frozen UA-ToolCall dataset and gold labels, function schemas, exact prompt construction, experimental protocol, model revisions and hashes, raw outputs, evaluation and

statistical-analysis code, and the executed analysis notebook – is publicly available at https://github.com/neocode/ua-toolcall. The protocol initially specified FP16 GPU inference; before results were interpreted, an execution addendum documented the change to a common Q4_K_M llama.cpp CPU stack. Tasks, gold labels, research questions, the primary endpoint, paired contrasts, error hierarchy, bootstrap seed, and multiplicity correction were unchanged.

Native tool APIs were excluded to avoid model-specific adapters. All models used the same instruction and JSON payload; chat templates were wrappers only. Each output was a single JSON object whose decision field was tool_call, clarify, or no_tool; no grammar or repair was used, and SmolLM3 used /no_think within the 128-token budget.

Inference used llama-cpp-python 0.3.34 with nine CPU threads on an AMD EPYC 9V74 processor, a context length of 8,192 tokens, temperature 0, seed 42, and at most 128 new tokens. Raw text and metadata were stored before parsing in resumable append-only JSONL; the final matrix had 1,200 unique records per model, with pilots excluded. Retained runs took 62.8, 53.1 (resume included), and 69.3 minutes; timing was not analysed. Table 3 lists the three builds and the final inference configuration. Exact model revisions, GGUF revisions, SHA-256 hashes, and raw-file integrity information corresponding to these runs are documented in the public UA-ToolCall repository (EXECUTION_ADDENDUM.md).

Table 3. **Evaluated models and final inference configuration**

Model	Parameters	Precision	Decoding
Qwen3-4B-Instruct-2507	4B	Q4_K_M	Greedy; 128 tokens
Granite-3.3-2B-Instruct	2B	Q4_K_M	Greedy; 128 tokens
SmolLM3-3B	3B	Q4_K_M	Greedy; 128 tokens; /no_think

Source: **compiled by the authors**

Evaluation metrics and statistical analysis. In this study, function-calling reliability is operationalized as the task-level strict-success rate under fixed inference settings. For positive tool-call cases, strict success is the primary outcome. It requires exactly one JSON object, decision tool_call, the correct function, schema-valid arguments, the gold field set, and canonical values. Markdown fences or surrounding prose fail strict JSON even when an object can be recovered from them, because the intended consumer is a program parser and not a human reader.

Diagnostics include strict and recoverable JSON validity, decision and selection accuracy, schema validity, exact arguments, per-field accuracy, clarification fields, and unnecessary calls. Component diagnostics are scored independently of earlier stages: schema validity is checked against the predicted tool when that tool name is known, with the gold-tool schema used only when no known tool name is available. Semantic success accepts only aliases prespecified before inference; it may recognize a meaning-preserving localized form but cannot convert it into operational success. The semantic–strict gap quantifies outputs that satisfy the prespecified semantic criteria but violate the machine contract.

Errors are assigned once, in this order: format, decision, tool selection, schema, semantic argument, canonicalization, other. Under this rule, гривня in place of the required enum value UAH counts as a schema failure, and an unknown tool name outranks any defect in its arguments.

The primary paired contrast is EN query + EN descriptions + full20 versus UK query + EN descriptions + full20. It uses risk differences, 95 % task-bootstrap intervals (10,000 draws; seed 20260721), and two-sided exact McNemar tests; three model p-values per contrast receive Holm correction. Secondary contrasts test Ukrainian descriptions, easy5 versus hard5, and bilingual descriptions. The catalogue contrasts are interpreted only for catalogue-order seed 1701.

The eight core conditions are also analysed by binomial generalized estimating equations over the 100 positive tasks per cell, clustered by task ID, with request language, description language, catalogue, their interactions, and fixed model indicators. The pooled GEE includes no model-by-

condition interactions; model-specific inference relies on the prespecified paired contrasts. Effect sizes, intervals, and discordant pairs take priority over binary significance.

The 100 positive pairs balance sensitivity and compute cost. With illustrative discordance 0.25 and directional imbalance 0.15, about 88 pairs provide 80% power at two-sided 0.05 by a normal approximation. This is design rationale, not a claim about the observed effect; the smaller negative and clarification subsets are diagnostic.

The 10 clarification tasks and 10 no-tool tasks are repeated across the 10 conditions, so the pooled 100 records per model are not treated as independent observations. Each condition is reported separately as successes/10 with Wilson 95 % intervals. Pooled no-tool rates are descriptive and receive 95 % task-ID-cluster bootstrap intervals obtained by resampling the 10 task IDs across all conditions (10,000 draws; seed 20260813).

Results. RQ1 – Query-language effect. Under English descriptions and full20, strict success fell from 94 % to 74 % for Qwen (paired difference -20 percentage points, 95% CI -29 to -12; Holm-adjusted $p = 7.18 \cdot 10^{-5}$), from 43 % to 9 % for Granite (-34 points,CI -44 to -24;$p = 5.84 \cdot 10^{-8}$), and from 26 % to 9 % for SmolLM3 (-17 points, CI -27 to -8; $p = 9.11 \cdot 10^{-4}$). Fig. 1 shows the condition rates with Wilson intervals. Qwen retained 100 % tool-selection accuracy in both languages, while exact arguments fell from 94 % to 74 %. Granite tool selection fell from 66% to 30% and exact arguments from 55 % to 12 %; SmolLM3 changed from 84 % to 77 % in selection and from 62 % to 26 % in exact arguments. In the pooled GEE, a Ukrainian query had OR 0.189 (95% CI 0.120-0.296, $p < 10^{-12}$).

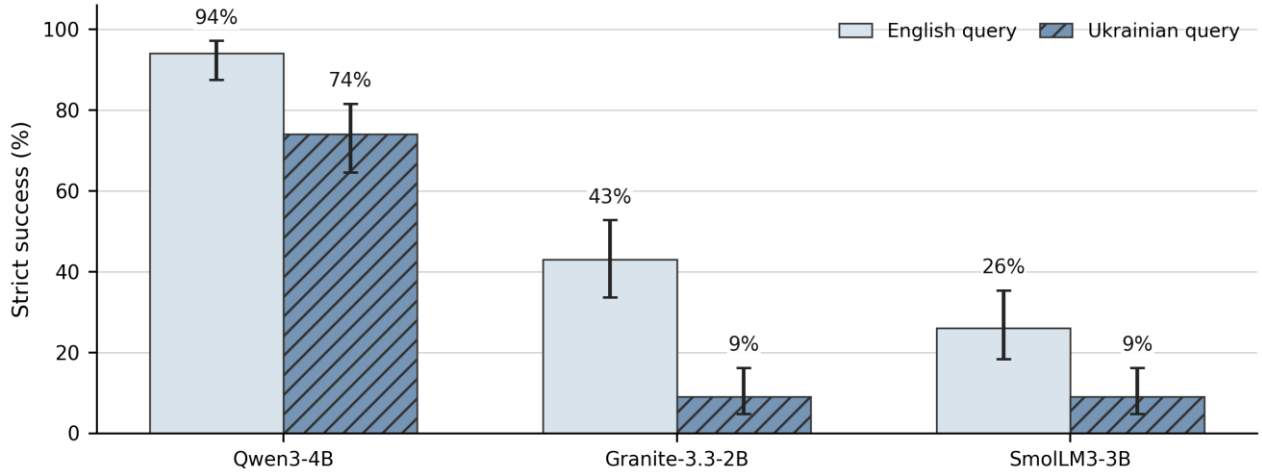

Fig. 1. **Strict success by query language under English descriptions and full20**
Source: **compiled by the authors**

RQ2 – Description localization. For Ukrainian/full20 requests, Ukrainian descriptions increased SmolLM3 strict success from 9 % to 20 % (+11 points, CI +5 to +18; 12 gains/1 loss; Holm p = 0.0103). Its exact-argument and semantic-success changes were +11 and +14 points. Qwen changed from 74 % to 76 % (+2 points, CI -5 to +9; Holm p = 1.0), and Granite from 9 % to 10 % (+1 point, CI -3 to +6; Holm p = 1.0). These intervals do not establish benefit or equivalence for Qwen or Granite. Table 4 also records the description switch outside the prespecified contrasts: under Ukrainian queries in hard5 it moved Qwen from 76 % to 77 %, Granite from 5 % to 8 % and SmolLM3 from 15 % to 11 %, and under English queries it moved Granite from 43 % to 23 % in full20 and SmolLM3 from 34 % to 23 % in hard5 but from 26 % to 39 % in full20. These cells were not part of the prespecified paired contrasts and carry no adjusted inference; descriptively, the direction of the change varied across catalogue conditions and models. The pooled Ukrainian-query × Ukrainian-description interaction was positive (OR 1.714, 95% CI 1.212-2.422, p = 0.0023), but the model-specific paired contrasts show that the interaction was not uniform.

RQ3 – Catalogue complexity. With Ukrainian queries and English descriptions, easy5-to-hard5 strict differences were 0 points for Qwen (76 % to 76 %; CI -5 to +5), +3 for Granite (2 % to

5%; CI -2 to +8), and +4 for SmolLM3 (11% to 15 %; CI -3 to +11); all Holm-adjusted p-values were 1.0. Hard5-to-full20 differences were -2 points for Qwen (CI -8 to +4), +4 for Granite (CI -3 to +11), and -6 for SmolLM3 (CI -14 to +2). After Holm correction neither the fixed-size distractor contrast nor the catalogue-size contrast reached significance. In hard5 the paired distractor accounted for 0 of 1 Qwen, 3 of 9 Granite, and 1 of 1 SmolLM3 selection failures. In full20 Qwen produced no selection failures at all, while the paired distractor accounted for 1 of 17 Granite failures and 1 of 1 for SmolLM3. These estimates describe only the recorded seed-1701 catalogue order; they do not estimate robustness to alternative positions or counterbalanced orders.

Table 4. **Strict success in all ten experimental conditions (95 % Wilson confidence intervals)**

Condition	Qwen3-4B	Granite-3.3-2B	SmolLM3-3B
EN query / EN descriptions, hard5	93 % (86.3-96.6)	32 % (23.7-41.7)	34 % (25.5-43.7)
EN query / UK descriptions, hard5	95 % (88.8-97.8)	34 % (25.5-43.7)	23 % (15.8-32.2)
UK query / EN descriptions, hard5	76 % (66.8-83.3)	5 % (2.2-11.2)	15 % (9.3-23.3)
UK query / UK descriptions, hard5	77 % (67.8-84.2)	8 % (4.1-15.0)	11 % (6.3-18.6)
EN query / EN descriptions, full20	94 % (87.5-97.2)	43 % (33.7-52.8)	26 % (18.4-35.4)
EN query / UK descriptions, full20	93 % (86.3-96.6)	23 % (15.8-32.2)	39 % (30.0-48.8)
UK query / EN descriptions, full20	74% (64.6-81.6)	9 % (4.8-16.2)	9 % (4.8-16.2)
UK query / UK descriptions, full20	76 % (66.8-83.3)	10 % (5.5-17.4)	20 % (13.3-28.9)
UK query / EN descriptions, easy5	76 % (66.8-83.3)	2 % (0.6-7.0)	11 % (6.3-18.6)
UK query / bilingual descriptions, full20	76 % (66.8-83.3)	14 % (8.5-22.1)	26 % (18.4-35.4)

Source: **compiled by the authors**

RQ4 – Bilingual mitigation. Under Ukrainian/full20, bilingual descriptions increased SmolLM3 strict success from 9 % to 26 % (+17 points, CI +9 to +26; 19 gains/2 losses; Holm $p = 6.64 \cdot 10^{-4}$). Qwen changed from 74 % to 76 % (+2 points, CI -3 to +8; Holm p = 0.727), and Granite from 9 % to 14 % (+5 points, CI -1 to +12; Holm p = 0.453). Fig. 2 contrasts these effects with Ukrainian-only descriptions. Mean prompt length rose from 2,366 to 4,154 tokens for Qwen (+75.5%), from 2,533 to 4,637 for Granite (+83.1 %), and from 2,374 to 3,652 for SmolLM3 (+53.8%). So the one statistically significant gain came at a substantial context-length cost, and it did not carry over to the other two models.

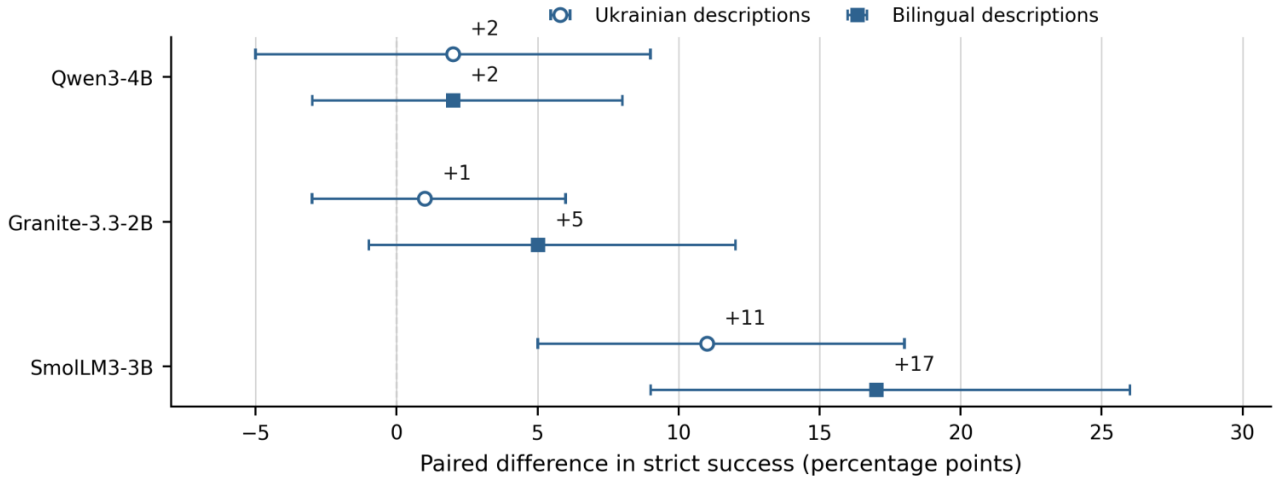

Fig. 2. **Paired changes in strict success relative to English descriptions under Ukrainian/full20**

Source: **compiled by the authors**

RQ5 – Error mechanism. Across 1,000 positive-condition outputs per model, strict/semantic success was 83.0 %/90.7 % for Qwen (gap 7.7 points), 18.0 %/28.8 % for Granite (gap 10.8), and 21.4 %/23.4 % for SmolLM3 (gap 2.0). Qwen's 170 strict failures were 52.4 % semantic-argument errors and 45.3 % canonicalization mismatches. Granite's 820 failures were led by decision (37.2%), format (27.8 %), and semantic-argument errors (19.8 %). SmolLM3's 786 failures were dominated by decision errors (74.0 %), followed by semantic-argument errors (17.0 %). Exact clarification success (strict JSON, the correct clarify decision, and the exact missing-field set) was 0/10 in every one of the 30 model-by-condition cells; each cell has a 95 % Wilson interval of 0–27.8 %. Because the same 10 tasks recur across conditions, the pooled 0/100 counts are descriptive only. Exact no-tool response success pooled over conditions was 47/100 for Qwen, 12/100 for SmolLM3, and 6/100 for Granite; the corresponding 95 % task-ID-cluster bootstrap intervals were 31-64 %, 1-27 %, and 0-12 %. Table 4 reports strict success in all ten positive-task conditions, Table 5 the diagnostic breakdown of the primary contrast, Table 6 the per-condition no-tool results, and Fig. 3 the hierarchical composition of strict failures.

Table 5. **Diagnostic outcomes in the primary language contrast (%)**

Metric	Qwen3-4B EN / UK	Granite-3.3-2B EN / UK	SmolLM3-3B EN / UK
Tool selection	100 / 100	66 / 30	84 / 77
Schema valid (component)	100 / 100	83 / 45	87 / 90
Exact arguments	94 / 74	55 / 12	62 / 26
Semantic success	97 / 88	43 / 11	26 / 10
Strict success	94 / 74	43 / 9	26 / 9

Source: **compiled by the authors**

Table 6. **Exact no-tool success by condition (successes/10 and 95 % Wilson confidence intervals)**

Condition	Qwen3-4B	Granite-3.3-2B	SmolLM3-3B
EN query / EN descriptions, hard5	9/10 (59.6-98.2)	2/10 (5.7-51.0)	3/10 (10.8-60.3)
EN query / UK descriptions, hard5	7/10 (39.7-89.2)	2/10 (5.7-51.0)	2/10 (5.7-51.0)
UK query / EN descriptions, hard5	1/10 (1.8-40.4)	0/10 (0.0-27.8)	2/10 (5.7-51.0)
UK query / UK descriptions, hard5	4/10 (16.8-68.7)	0/10 (0.0-27.8)	1/10 (1.8-40.4)
EN query / EN descriptions, full20	9/10 (59.6-98.2)	2/10 (5.7-51.0)	2/10 (5.7-51.0)
EN query / UK descriptions, full20	8/10 (49.0-94.3)	0/10 (0.0-27.8)	1/10 (1.8-40.4)
UK query / EN descriptions, full20	1/10 (1.8-40.4)	0/10 (0.0-27.8)	0/10 (0.0-27.8)
UK query / UK descriptions, full20	3/10 (10.8-60.3)	0/10 (0.0-27.8)	1/10 (1.8-40.4)
UK query / EN descriptions, easy5	2/10 (5.7-51.0)	0/10 (0.0-27.8)	0/10 (0.0-27.8)
UK query / bilingual descriptions, full20	3/10 (10.8-60.3)	0/10 (0.0-27.8)	0/10 (0.0-27.8)

Source: compiled by the authors

Error analysis and discussion. The observed language effect arose from different error mechanisms across models. Qwen made no selection errors in either primary condition but emitted recognizable non-canonical values: Odessa where the contract required Odesa, and Uzhgorod instead of Uzhhorod. Odessa belonged to the alias set prespecified before inference, so it counted as a semantic success and an operational failure; Uzhgorod was not on that list and failed both criteria. MLCL reports the same execution-level mismatch [5].

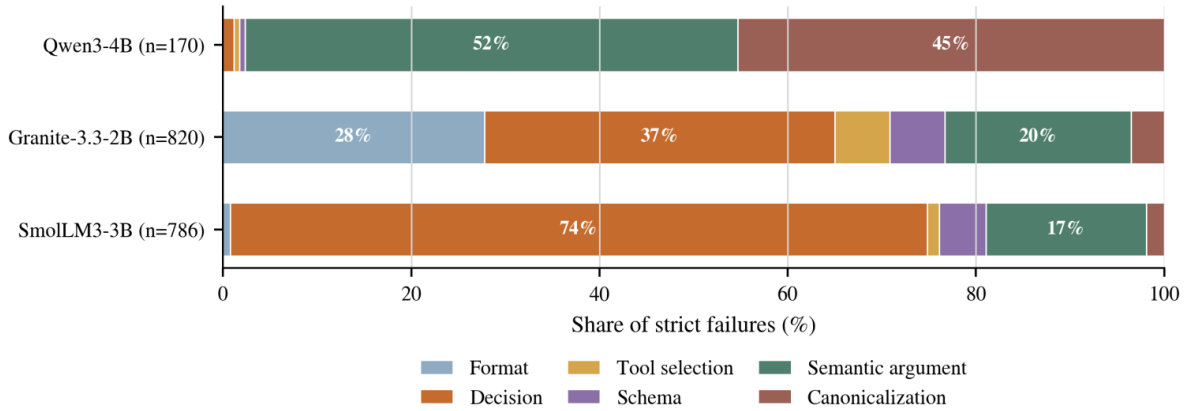

Fig. 3. **Hierarchical composition of strict failures across positive conditions**
Source: **compiled by the authors**

Granite and SmolLM3 failed earlier. Granite sometimes translated the request instead of calling weather_current or requested clarification for inputs that were already complete. SmolLM3 often placed the tool name in the decision field, for example “decision”:”weather_current”, violating the fixed three-value contract. Its small semantic–strict gap comes from decision errors, not from recoverable locale variants. A constrained wrapper could change this result and should be evaluated separately [21].

Localization was model-specific. SmolLM3 benefited, Qwen had little selection headroom, and Granite’s bilingual selection gain was offset by later-stage errors. Bilingual prompts grew by 54-83%, so any accuracy benefit carries a substantial context-length cost. The non-significant catalogue contrasts do not establish robustness: their intervals permit moderate effects, and Granite’s full-catalogue failures spread to unrelated tools. Earlier reports that minor wording changes and similar distractors destabilize tool preferences point the same way [19], [20].

Deployment should validate the decision enum, tool name, schema, and canonical arguments before execution; normalize only aliases explicitly accepted by the backend; and test clarification and no-tool behaviour separately. Description language should be selected from paired evidence. In these experiments, only SmolLM3 showed a statistically significant strict-success improvement with bilingual descriptions. Zero exact clarification success also motivates explicit clarification training or constrained policy logic [2].

Threats to validity. The unified prompt protocol standardizes the interface across models but does not evaluate native tool-calling adapters. Strict scoring is intentionally demanding; constrained decoding or approved normalization may improve production reliability. Prespecified aliases may under-credit valid paraphrases. Equal Q4_K_M quantization and llama.cpp improve within-study comparability but limit comparison with FP16 or vendor results.

Internal validity depends on bilingual equivalence, catalogue construction, and ordering. Schema checks and a post-hoc side-by-side audit of all 144 English-Ukrainian pairs found no mismatch requiring correction; however, the audit was not completed before inference. Semantic confusability was defined by the catalogue design rather than independently similarity-rated. One seed-fixed order removes within-pair position differences but does not estimate order variability, so RQ3 is configuration-specific. Cache reuse and resume changed only execution cost; outputs were append-only and scored after all files reached 1,200 unique records. Pilots were excluded.

External validity is limited to 20 office/service functions, one-step JSON-object generation from a shared text prompt, three compact Q4_K_M model builds, llama.cpp CPU inference, greedy unconstrained decoding, one catalogue order, and one reference time. Native function-calling APIs, constrained decoding, multi-step interaction, and external side effects were not tested. Ten clarification and ten no-tool tasks are diagnostic and repeated across conditions, so the findings concern the recorded configurations rather than Ukrainian-language function calling or LLMs in general.

Conclusions. UA-ToolCall provides a controlled test of Ukrainian structured function calling. Under English descriptions and full20, Ukrainian queries reduced strict success by 20 points for

Qwen, 34 for Granite, and 17 for SmolLM3; every paired effect survived Holm correction. Qwen kept 100 % selection accuracy and lost arguments/canonical forms; Granite lost selection and arguments; SmolLM3 was dominated by decision-field errors. Ukrainian and bilingual descriptions produced statistically significant gains of 11 and 17 points for SmolLM3, while bilingual prompts grew by 54-83 %; no statistically significant gain appeared for the other models. Within the single catalogue order generated with seed 1701, catalogue contrasts were not statistically significant and established neither equivalence nor order-robustness. Across the positive conditions, strict and semantic success were 83.0 % and 90.7 % for Qwen, 18.0 % and 28.8 % for Granite, and 21.4 % and 23.4 % for SmolLM3, leaving semantic–strict gaps of 7.7, 10.8, and 2.0 points. Read against the prespecified directional hypotheses, the query-language and error-mechanism predictions (H1, H5) were borne out, the two localization predictions (H2, H4) held for SmolLM3 alone, and the catalogue prediction (H3) was not supported. For deployment this means validating the decision field, the schema, and the canonical argument values before execution, and choosing the description language separately for each model.

Conflict of interests: The authors declare no financial, personal, authorship, or other conflicts of interest that could have influenced the research or the results reported in this article

Financing: The research was carried out without financial support

Accessibility data: The complete reproducibility package supporting this study is publicly available at https://github.com/neocode/ua-toolcall.

Using artificial intelligence tools: The authors confirm that they did not use instrumental AI tools for writing this one work.

REFERENCES

1. Patil, S. G., et al. “The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models”. *Proceedings of the 42nd International Conference on Machine Learning*. 2025; 267: 48371–48392. – Available from: https://proceedings.mlr.press/v267/patil25a.html. – [Accessed: 12 March 2026].

2. Ross, H., Mahabaleshwarkar, A. S. & Suhara, Y. “When2Call: When (not) to call tools”. *Proceedings of the Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2025; 1: 3391–3409. DOI: https://doi.org/10.18653/v1/2025.naacl-long.174.

3. Lu, J., et al. “ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities”. *Findings of the Association for Computational Linguistics:* 2025. p. 1160–1183. DOI: https://doi.org/10.18653/v1/2025.findings-naacl.65.

4. Kulkarni, M., et al. “MASSIVE-Agents: A benchmark for multilingual function-calling in 52 languages”. *Findings of the Association for Computational Linguistics*. 2025 p. 20193–20215. DOI: https://doi.org/10.18653/v1/2025.findings-emnlp.1099.

5. Luo, Z., et al. “Lost in execution: On the multilingual robustness of tool calling in large language models”. *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*. 2026; 1: 44059–44077. DOI: https://doi.org/10.18653/v1/2026.acl-long.2039.

6. Kravchenko, A., Paniv, Y. & Drushchak, N. “UAlign: LLM alignment benchmark for the Ukrainian language”. *Proceedings of the Fourth Ukrainian Natural Language Processing Workshop*. 2025. p. 36–44. DOI: https://doi.org/10.18653/v1/2025.unlp-1.4.

7. Syromiatnikov, M. V. & Ruvinskaya, V. M. “UA-Code-Bench: A competitive programming benchmark for evaluating large language models code generation in Ukrainian”. *Informatics. Culture. Technology*. 2025; 2 (2): 308–314. DOI: https://doi.org/10.15276/ict.02.2025.47.

8. Paniv, Y. “Isolating LLM performance gains in pre-training versus instruction-tuning for mid-resource languages: The Ukrainian benchmark study”. *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing – Natural Language Processing in the Generative AI Era*. 2025 p. 876–883. DOI: https://doi.org/10.26615/978-954-452-098-4-100.

9. Ye, J., et al. “RoTBench: A multi-level benchmark for evaluating the robustness of large language models in tool learning”. *Proceedings of the 2024 Conference on Empirical Methods in*

Natural Language Processing. 2024. p. 313–333. DOI: https://doi.org/10.18653/v1/2024.emnlp-main.19.

10. Guo, Z., et al. "StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models". *Findings of the Association for Computational Linguistics*. 2024. p. 11143–11156. DOI: https://doi.org/10.18653/v1/2024.findings-acl.664.

11. Wang, J., et al. "HammerBench: Fine-grained function-calling evaluation in real mobile device scenarios". *arXiv*. 2025. DOI: https://doi.org/10.48550/arXiv.2412.16516.

12. Chen, C., et al. "ACEBench: A Comprehensive evaluation of LLM tool usage". *Findings of the Association for Computational Linguistics*. 2025. p. 12970–12998. DOI: https://doi.org/10.18653/v1/2025.findings-emnlp.697.

13. Liu, Z., et al. "APIGen: Automated PIpeline for generating verifiable and diverse function-calling datasets". *Advances in Neural Information Processing Systems*. 2024; 37: 54463–54482. DOI: https://doi.org/10.52202/079017-1725.

14. Liu, W., et al. "ToolACE: Winning the points of LLM function calling". *The Thirteenth International Conference on Learning Representations.* 2025. – Available from: https://openreview.net/forum?id=8EB8k6DdCU. – [Accessed: 16 March 2026].

15. Hofman O., et al. "MAPS: A multilingual benchmark for agent performance and security". *Findings of the Association for Computational Linguistics*. 2026. p. 821–845. DOI: https://doi.org/10.18653/v1/2026.findings-eacl.42.

16. Chen, Y.-C. et al. "Enhancing function-calling capabilities in LLMs: Strategies for prompt formats, data integration, and multilingual translation". *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2025; 3: 99–111. DOI: https://doi.org/10.18653/v1/2025.naacl-industry.9.

17. Ersoy, A. et al. "Tool calling for arabic LLMs: Data strategies and instruction tuning". *Proceedings of the Third Arabic Natural Language Processing Conference*. 2025. p. 347–358. DOI: https://doi.org/10.18653/v1/2025.arabicnlp-main.28.

18. Emanuilov, S. "Teaching a language model to speak the language of tools". *arXiv*. 2025. DOI: https://doi.org/10.48550/arXiv.2506.23394.

19. Faghih, K., et al. "Tool preferences in agentic LLMs are unreliable". *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 2025. p. 20954–20969. DOI: https://doi.org/10.18653/v1/2025.emnlp-main.1060.

20. Rabinovich, E. & Anaby Tavor, A. "On the robustness of agentic function calling". *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP*). 2025. p. 298–304. DOI: https://doi.org/10.18653/v1/2025.trustnlp-main.20.

21. Geng, S., et al. "JSONSchemaBench: A rigorous benchmark of structured outputs for language models". *arXiv*. 2025. DOI: https://doi.org/10.48550/arXiv.2501.10868.

22. Qin, Y., et al. "Tool learning with foundation models". *ACM Computing Surveys*. 2024; 57 (4): 3704435, https://www.scopus.com/pages/publications/85213123636. DOI: https://doi.org/10.1145/3704435.

Received 14.07.2026
Received after revision 28.08.2026
Accepted 03.09.2026

DOI: https://doi.org/10.15276/ict.03.2026.18
УДК 004.8:004.4

Оцінювання надійності виклику функцій великими мовними моделями для україномовних запитів

Шеремет Олексій Іванович[1)]
ORCID: https://orcid.org/0000-0003-1298-3617; oleksii.sheremet@ddma.edu.ua. Scopus Author ID: 57170410800
Томашевський Роман Сергійович[1)]
ORCID: https://orcid.org/0000-0002-5278-9272; roman.tomashevskyi@ddma.edu.ua. Scopus Author ID: 56338488300

Чмихова Оксана Володимирівна[1)]
ORCID: https://orcid.org/0000-0002-9198-9701; oksana.chmykhova@ddma.edu.ua. Scopus Author ID: 57194619495
Воробйов Богдан Віталійович[1)]
ORCID: https://orcid.org/0000-0002-0264-354X; bohdan.vorobiov@ddma.edu.ua. Scopus Author ID: 57891861500
[1)] Донбаська державна машинобудівна академія, вул. Академічна, 72. Краматорськ, 84313, Україна

АНОТАЦІЯ

Виклик функцій великими мовними моделями дедалі частіше використовують для перетворення природномовних запитів на виклики програмних функцій, однак надійність цього процесу для україномовних запитів залишається недостатньо дослідженою. Запропоновано UA-ToolCall – парний англо-український еталонний набір із двадцятьма двомовними контрактами імітаційних функцій, двадцятьма чотирма прикладами для налагодження і ста двадцятьма зафіксованими тестовими завданнями: ста завданнями з обов'язковим викликом інструмента, десятьма завданнями на уточнення та десятьма завданнями без виклику інструмента. У ньому варіюються мова запиту, мова описів, розмір каталогу та подібність інструментів-дистракторів, тоді як машинні ідентифікатори залишаються канонічними. Моделі сімейств Qwen3, Granite 3.3 і SmolLM3 оцінено у форматі Q4_K_M за спільним текстовим промптом для одноетапного формування структурованих результатів виклику функцій без обмеженого декодування; застосовано жадібне декодування й отримано три тисячі шістсот відповідей. За англомовних описів і повного каталогу з двадцяти інструментів українські запити знизили строгу успішність із дев'яноста чотирьох до сімдесяти чотирьох відсотків, із сорока трьох до дев'яти відсотків і з двадцяти шести до дев'яти відсотків відповідно; усі парні ефекти залишилися значущими після поправки Холма. Qwen зберіг стовідсоткову точність вибору інструмента, тому його втрати були пов'язані з аргументами й канонізацією значень; Granite втрачав і вибір, і аргументи, а в SmolLM3 переважали помилки в полі рішення. Українські та двомовні описи підвищили строгу успішність SmolLM3 на одинадцять і сімнадцять відсоткових пунктів відповідно, але не дали статистично значущого приросту іншим моделям; двомовні описи збільшили середню довжину промпту на п'ятдесят чотири – вісімдесят три відсотки. За єдиного порядку каталогу, сформованого з початковим значенням 1701, жоден контраст каталогу не був статистично значущим; варіативність, зумовлену порядком інструментів, не оцінювали. Результати підтверджують доцільність окремого оцінювання семантичної та виконуваної правильності й показують, що користь від локалізації залежить від моделі. Висновки стосуються одноетапного формування структурованих результатів виклику функцій за текстовим промптом у досліджених квантизованих моделях, а не штатних інтерфейсів виклику функцій, обмеженого декодування чи багатокрокової взаємодії з інструментами.

Ключові слова: великі мовні моделі; виклик функцій; використання інструментів; багатомовне оцінювання; структуроване виведення

ABOUT THE AUTHORS

Oleksii I. Sheremet - Doctor of Engineering Sciences, Professor, Head of Department of Electrical Engineering and Renewable Energy. Donbas State Engineering Academy, 72, Akademichna St. Kramatorsk, 84313, Ukraine
ORCID: https://orcid.org/0000-0003-1298-3617, oleksii.sheremet@ddma.edu.ua. Scopus Author ID: 57170410800
Research field: Artificial intelligence, automatic control systems, electromechanical systems

Шеремет Олексій Іванович - доктор технічних наук, професор, завідувач кафедри Електроінженерії та відновлювальної енергетики. Донбаська державної машинобудівна академія, вул. Академічна, 72. Краматорськ, 84313, Україна
***Галузь досліджень*:** Штучний інтелект, системи автоматичного керування, електромеханічні системи

Roman S. Tomashevskyi - Doctor of Engineering Sciences, Professor, Acting Rector, Donbas State Engineering Academy, 72 Akademichna St. Kramatorsk, 84313, Ukraine
ORCID: https://orcid.org/0000-0002-5278-9272; roman.tomashevskyi@ddma.edu.ua. Scopus Author ID: 56338488300
Research field: Biomedical electronics, information and measurement systems, water-treatment systems

Томашевський Роман Сергійович - доктор технічних наук, професор, в. о. ректора. Донбаська державна машинобудівна академія, вул. Академічна, 72. Краматорськ, 84313, Україна
Галузь досліджень: Біомедична електроніка, інформаційно-вимірювальні системи, системи водоочищення.

Oksana V. Chmykhova - Candidate of Engineering Sciences, Associate Professor, First Vice-Rector and Vice-Rector for Academic and Methodological Work, Donbas State Engineering Academy, 72, Akademichna St. Kramatorsk, 84313, Ukraine, ORCID: https://orcid.org/0000-0002-9198-9701, oksana.chmykhova@ddma.edu.ua. Scopus Author ID: 57194619495
Research field: Biomedical electronics, sensor systems, power electronics.

Чмихова Оксана Володимирівна - кандидат технічних наук, доцент, перша проректорка та проректорка з науково-педагогічної та методичної роботи. Донбаська державна машинобудівна академія, вул. Академічна, 72. Краматорськ, 84313, Україна
Галузь досліджень: Біомедична електроніка, сенсорні системи, силова електроніка

Bohdan V. Vorobiov - PhD, Associate Professor, Vice-Rector for Research, Development Management and International Relations, Donbas State Engineering Academy, 72, Akademichna St. Kramatorsk, 84313, Ukraine
ORCID: https://orcid.org/0000-0002-0264-354X, bohdan.vorobiov@ddma.edu.ua. Scopus Author ID: 57891861500
Research field: Electromechanical systems, neural-network control, renewable energy

Воробйов Богдан Віталійович - доктор філософії (PhD), доцент, проректор з наукової роботи, управління розвитком та міжнародних зв'язків. Донбаська державна машинобудівна академія, вул. Академічна, 72. Краматорськ, 84313, Україна,
Галузь досліджень: Електромеханічні системи, нейромережеве керування, відновлювана енергетика