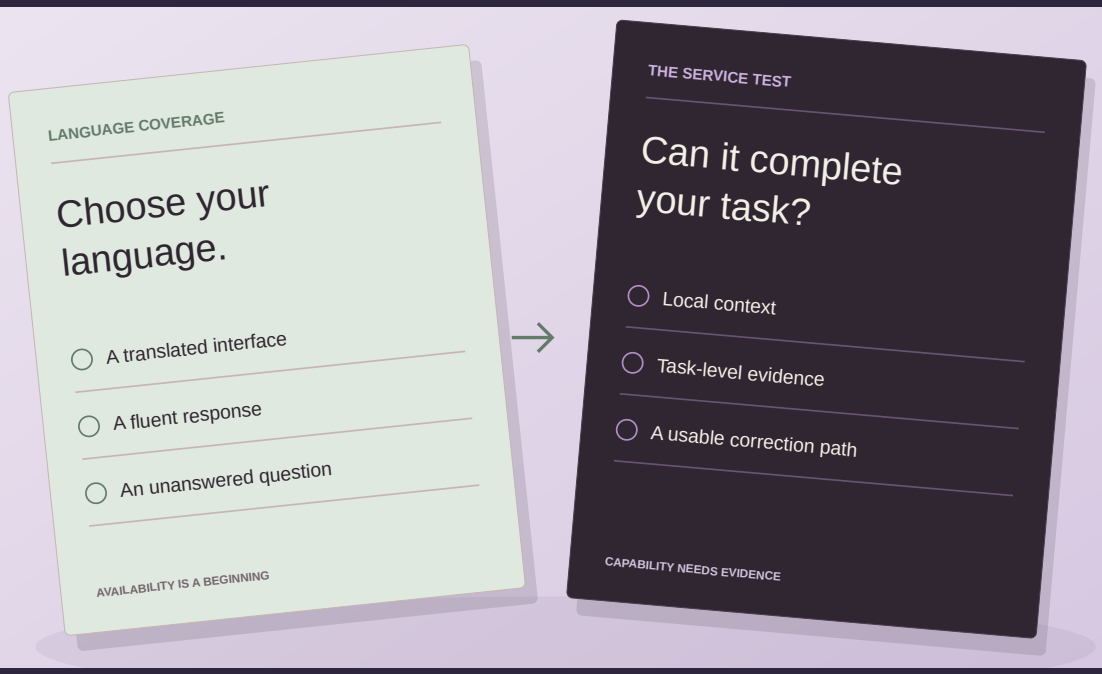
kymata labs

RESEARCH PAPER 05 / AI & LANGUAGE ACCESS

The AI That Doesn't Speak Your Language

A dropdown can open the door. It cannot stand behind the answer.



Kymata Labs Research
2026-11-16 · Edition 1.0

Independent analysis
kymatalabs.com

Small studio. Open worlds.

THE LONGER VIEW

Executive summary.

A multilingual interface can make access visible while leaving capability and accountability unproven. This paper examines the task definitions and exclusions in XTREME, FLORES-200, TyDi QA, Belebele, PAWS-X and MLQA. The crucial distinction is between accepting a language, demonstrating a specific capability and repairing a consequential mistake. TyDi QA provides a revealing example: its simplified GoldP task removes unanswerable questions and two languages for tooling compatibility, changes that must travel with any reported result. We develop a service-level account of translation, retrieval, answerability and recourse, then propose a support contract and a worked municipal-service release exercise. The strongest counterargument matters: demanding perfect parity can exclude people from useful tools. The alternative is a calibrated promise with evidence and a remedy, not an English-only gate. No current model ranking or measured present-day language disparity is claimed.

> Analysis of original benchmark repository documentation and released prompting notes, captured November 16, 2026. We inspected task definitions, language coverage, exclusions and evaluation guidance; we did not read the full linked papers, rerun benchmarks, audit all data, test current commercial models or interview users. Historical release counts are not current capability measurements. The service contract, diagrams and municipal example are explicitly proposed analytical tools.

01 / The dropdown makes a promise
02 / A benchmark has borders
03 / The missing answer disappears before the test
04 / Translation solves one problem at a time
05 / The user should not have to become the translator of last resort
06 / Same language, different test
07 / The case for opening the door anyway
08 / Write a support contract that can fail
09 / Do not average away the person at the edge
10 / A service speaks through its remedies
11 / Evidence and method
Q&A / Questions worth asking
Sources / References

© 2026 Kymata Labs LLC. All rights reserved.
Read the web edition: kymatalabs.com/research/the-ai-that-doesnt-speak-your-language

The dropdown makes a promise

A language menu is a small piece of software with an unusually large moral vocabulary. It says welcome. It says you can ask your question here. In a product that answers questions, it can also suggest something the engineering team has never actually established: that the answer will deserve the same confidence in every language on the list.

The difference is easy to miss. A system can display the right script, understand a greeting and produce an elegant paragraph while remaining unevaluated for the decision the paragraph helps someone make. The interface works. The service promise is still an open question.

This paper is about that gap. It is not a claim that AI fails outside English, that English answers are dependable, or that a particular language community should wait for permission to use new technology. The original benchmark documentation inspected here cannot support any of those conclusions. It supports a more specific challenge: **a language listed is not a capability demonstrated, and a capability demonstrated is not a remedy delivered.**

Multilingual AI becomes an accountability problem when a provider lets those three things blur. A language selector makes access visible. A benchmark makes some performance measurable. A correction route makes mistakes contestable. Shipping the first does not discharge responsibility for the other two.

The commercial temptation is to count languages because languages are countable. The product question is harder: which people can accomplish which tasks, with which sources, under which conditions, and what happens when the system gets them wrong? That question will never fit into a dropdown. It belongs in the release decision.

A benchmark has borders

Consider XTREME, the 2020 benchmark for cross-lingual generalization. Its original documentation describes 40 languages and nine tasks, including classification, tagging, retrieval and question answering. Each task covers a subset of the benchmark's languages. The standard transfer setting fine-tunes on English task data and evaluates the same task in other languages. The BUCC and Tatoeba retrieval tasks instead use pretrained representations directly. [1]

Those details are the meaning of the result. Forty languages does not mean nine independently verified capabilities in each of forty languages. Nor does a result from that experimental setting measure a contemporary assistant's ability to explain a school notice, contest a bill or recognize that it lacks the relevant information. XTREME is useful research precisely because its boundaries are specified.

FLORES-200 addresses another problem: multilingual machine translation. Its 2022 release documentation describes 3,001 sentences taken from 842 distinct web articles, divided into development, development-test and hidden test splits. Some translations originated from Spanish, French, Russian or Modern Standard Arabic rather than English. The release also includes two script alternatives for four languages. [2]

That is a substantial evaluation resource. It does not make sentence translation interchangeable with local advice, document retrieval or complaint handling. A correctly translated answer can preserve an original error perfectly. An accurate translation benchmark can tell us something valuable without telling us whether a product will choose the right authority for a question.

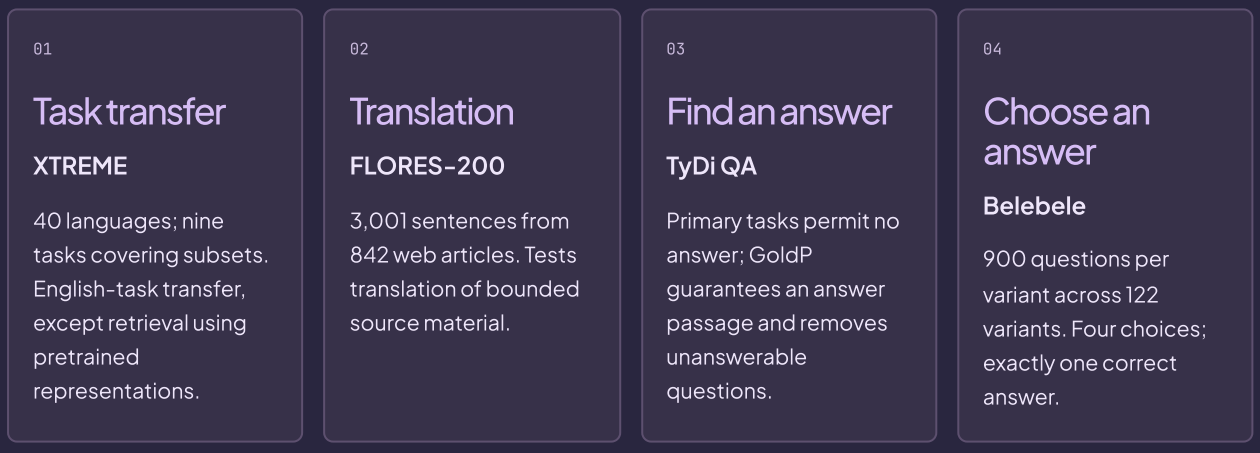
Belebele measures another slice. Its documentation describes 900 multiple-choice questions per language variant, covering 122 variants and 488 passages drawn from FLORES-200. Every question has four choices and exactly one correct answer. The parallel structure enables comparison on matched reading material. The same structure also defines the boundary: selecting among supplied alternatives is not open-ended assistance with missing or conflicting evidence. [4]

The numbers should therefore sit beside their tasks, not compete as marketing totals. A larger language count can mean wider coverage of a narrower problem. A smaller collection can ask a more demanding question about uncertainty. Neither establishes a universal hierarchy of usefulness.

FIGURE 01 HISTORICAL BENCHMARK SCOPE · NOT MODEL SCORES

A language count needs a task beside it.

Four evaluation resources ask different questions. Coverage counts do not measure interchangeable capabilities.



Original release definitions, principally 2020–2024. No current model ranking, language-parity estimate or country comparison is shown.

Original XTREME, FLORES-200, TyDi QA and Belebele repository documentation. [1] [2] [3] [4]

The missing answer disappears before the test

TyDi QA supplies an especially revealing example. The 2020 dataset describes 204,000 question-answer pairs across eleven typologically diverse languages. Questions were collected directly in each language from people who wanted to know an answer but did not yet know it, rather than translated from an English question set. [3]

Its primary tasks include finding a relevant passage in an article and identifying a minimal answer. Crucially, the valid outcomes include no answer. The system must sometimes determine that the supplied material does not resolve the question.

The same project offers a simplified task, TyDiQA-GoldP. Here the passage is guaranteed to contain an answer. Unanswerable questions are discarded. Thai and Japanese are removed because the absence of whitespace breaks some tools. The documentation explains these choices as a route to compatibility with existing question-answering pipelines, and explicitly asks researchers to name the particular task rather than report an undifferentiated “TyDi QA” result. [3]

There is nothing clandestine about this. The authors document the tradeoff. The problem begins if a buyer, vendor or product team discards the task descriptor while retaining the prestige of the benchmark. A test in which the answer is guaranteed cannot establish when an assistant should refuse to invent one. A tooling accommodation that removes languages cannot become evidence about the languages it removed.

This is the quiet way inequality can enter a technical system. It need not begin with a hostile policy. It can begin with a convenient file format, a parser assumption or a task chosen because the existing pipeline accepts it. Once the resulting score travels through a procurement slide, the missing cases become difficult to see.

The lesson is not that simplification is illegitimate. Research needs controlled tasks. It is that the exclusions must travel with the result. A model buyer should ask what the benchmark made easy before asking how high the score was.

Translation solves one problem at a time

A reasonable response is to translate the user's request into a language the system handles well, obtain an answer there and translate it back. That can be a valuable design. It also creates a chain that needs evaluation as a chain.

The first translation must preserve the user's meaning, including uncertainty, negation, names and conditions. Retrieval must then find evidence relevant to the actual setting. The answer must remain inside that evidence. The return translation must preserve its qualifications. Finally, the user must be able to question the outcome without being forced to restate the whole problem in another language.

These are analytical failure possibilities, not measured failure rates from the inspected datasets. They explain why fluency at the end cannot certify every step before it. A beautifully rendered answer can have crossed the wrong bridge at retrieval.

Imagine an explicitly hypothetical municipal-service assistant. A resident asks whether a document will be accepted for an application. The system translates the request competently, retrieves guidance from a similarly named program elsewhere and returns a clear answer in the resident's language. The failure is jurisdiction and source selection, not necessarily translation. Improving sentence-level translation alone need not repair it.

The same distinction matters in evaluation. MLQA is an extractive question-answering resource in seven languages. Its file naming separates the language of the context from the language of the question, allowing a question in one language to be paired with evidence in another. The documentation also requires the answer language when running the evaluator. [6]

A product that accepts a question in one language but searches only another has made a consequential architectural choice. That choice may be justified by the available sources. It should be visible, tested and revisable. Otherwise, the language selector promises a local conversation while the retrieval system silently supplies a different information environment.

FIGURE 02 PROPOSED SERVICE ANALYSIS · NOT MEASURED ERROR RATES

Fluency cannot certify the whole journey.

A multilingual answer depends on several distinct commitments. A polished final sentence does not verify the earlier steps.

01	02	03	04
Request	**Evidence**	**Answer**	**Recourse**
Preserve intent	**Find authority**	**Keep the limits**	**Make repair usable**
Keep names, conditions, uncertainty and negation intact across language handling.	Retrieve the relevant source for the actual question and setting; preserve its identity.	Stay within the evidence, preserve qualifications and recognize when no answer is supported.	Retain the exchange, route the dispute and communicate a correction the user can use.

Kymata Labs analytical framework. Source research motivates the task distinctions; it does not measure complaint handling or failures at these transitions.

Kymata Labs synthesis of translation, cross-language QA and answerability task definitions. [2] [3] [6]

The user should not have to become the translator of last resort

An institution can offer multilingual answers while keeping its accountability machinery effectively monolingual. Consider the general design problem, not a reported incident: the assistant speaks to the user in one language, but reporting an error requires an English form, an English explanation and evidence framed in terms an English-speaking reviewer recognizes.

The resulting burden is not simply another translation task. The person must first notice the defect, then explain why it matters, then persuade someone outside the original exchange that a correction is warranted. A fluent answer can make the first step harder to recognize; an inaccessible complaints

process can make the remaining steps impractical.

These mechanisms are not measured in the benchmark repositories. They are product responsibilities the benchmarks leave unanswered. Treating that absence as evidence of no problem would be as careless as treating it as evidence of universal failure.

A credible support claim should therefore include the route back. Can the user preserve the original question and answer? Can they flag the passage they believe is wrong? Will a reviewer see the source material and language setting that produced it? Can the institution communicate the correction in a form the user can use?

None of this requires promising a human expert instantly for every casual interaction. It requires matching the remedy to the consequence. An experimental writing aid and a service that influences access to a benefit should not inherit the same accountability standard merely because both display chat bubbles.

The distinction from subscription inequality matters. A free system can still transfer verification and translation work onto a user. A paid system can still lack a correction route for the language it accepts. The problem here is what the service can substantiate and repair, not only who can afford entry.

Same language, different test

Even the phrase “supports a language” can hide choices about script, variety, register and task. FLORES-200's language table distinguishes language-script combinations and named varieties. Belebele distinguishes 122 language variants from 115 distinct languages when scripts are ignored, and includes romanized variants in its documented inventory. These are dataset definitions, not a map of every way people communicate. [2][4]

An evaluation conducted in a formal written variety does not, by itself, establish behavior with informal spelling, mixed-language messages, speech transcription or a locally specific administrative term. Those are separate test conditions. The appropriate response is to state the tested scope and collect relevant examples, rather than label an entire population well served or poorly served.

There is also a difference between a matching string and a matching meaning. PAWS-X was built for paraphrase identification using translated sentence pairs. Its documentation distinguishes human-translated evaluation pairs from machine-translated training pairs. It also warns that individual first sentences can appear in both development and test data even though complete sentence pairs do not overlap. [5]

That caveat is valuable because it makes the evaluation unit explicit. Counting distinct sentences, distinct pairs and distinct decisions are not the same operation. A product team needs comparable care when it builds a local test set: repeated templates can inflate apparent variety, while a superficially new wording may probe exactly the same behavior.

Prompting and answer extraction matter too. Belebele's released instruction notes describe prompts asking models to emit a choice letter, plus processing that accepts forms such as a parenthesized letter. They also describe small instruction-format differences across the evaluated models. [7]

Those choices are reasonable, but they must be recorded. An observed gap can reflect task comprehension, instruction handling, output formatting or several things together. A responsible report distinguishes failures to choose correctly from failures to produce the expected format. It does not quietly fix one language's outputs more generously than another's.

The case for opening the door anyway

There is a strong objection to demanding better evidence: the demand can become another gate. If multilingual products must achieve certainty before launch, while English products are allowed to improve in public, an accountability argument will reproduce the exclusion it criticizes.

That objection is right. Perfect parity is neither established by the evidence here nor a reasonable prerequisite for every use. A tool that helps someone draft a letter, understand a difficult paragraph or begin research may provide substantial value even with clear limitations. The alternative may be a slower, more expensive or entirely unavailable service.

Users also have agency. It would be paternalistic to assume that a speaker cannot judge an imperfect tool, or that a provider outside the community should decide that no access is preferable. A language benchmark score alone cannot settle that choice.

The answer is a calibrated promise. Make useful low-consequence capabilities available with specific limits. Distinguish assisted drafting from authoritative guidance. Let users see the source when the task depends on one. Escalate cases whose consequences exceed the evidence available. Improve the service with speakers who can evaluate its practical usefulness, rather than treating them solely as test subjects.

This approach carries costs. Local evaluation, reviewer time and maintained source collections are work. Small teams cannot cover every combination at once. But a narrower, honest support statement is more useful than a broad claim whose exceptions appear only after a person relies on it.

There is a second counterargument: benchmark research is moving, so older datasets cannot describe current systems. That is also correct. This paper uses historical resources to inspect the logic of evaluation, not to rank today's models. Better contemporary models can improve outcomes substantially without dissolving the distinction between a translation task, a reading task and an accountable service.

The burden should fall on the claim. If a provider claims only experimental conversation, the evidence obligation differs from a claim of dependable local advice. The point is not to close the door. It is to stop putting a universal warranty above it.

Write a support contract that can fail

A useful multilingual release record is specific enough to be disproved. “Our assistant supports Spanish” is too broad to guide an evaluation. An illustrative alternative is: “This version summarizes the named service documents, in the tested written varieties, without deciding eligibility; unsupported or conflicting

cases are routed to a reviewer." That is a proposed contract, not a verified product claim.

The record should name the task, evidence collection, tested language conditions, model and retrieval versions, decision limits, and correction route. It should also identify what has not been tested. A missing evaluation is neither a pass nor a measured failure.

Start with actual user purposes, collected with appropriate permission and privacy protection. A source-grounded assistant needs examples with a clear answer, examples whose necessary evidence is absent, and examples where documents disagree. Have qualified reviewers define acceptable behavior before looking at the model's response. Translation checking and domain checking may require different people.

Use both matched cases and locally originated cases. Matched cases help compare behavior across languages on similar material. Locally originated cases help reveal questions and assumptions that a translated English inventory may never contain. TyDi QA's direct collection and Belebele's parallel design illustrate why these are complementary tools rather than competing claims to authenticity. [3][4]

Record the complete journey. Did the system retrieve the right source? Did the answer preserve its conditions? Did it decline when the evidence was insufficient? Did the user receive a usable next step? Could a reviewer reconstruct and correct the exchange? Keep these outcomes separate before deciding whether an aggregate number is useful.

Choose acceptance rules before release and tie them to consequence. Some defects may justify restricting a feature rather than blocking the language entirely. A system that can safely summarize documents but cannot adjudicate conflicts should retain the first function while clearly routing the second. Restrictions should have owners and review dates so that temporary limits do not become permanent neglect.

FIGURE 03

PROPOSED RELEASE CONTRACT

Publish the promise with its proof and remedy.

A useful support statement names a bounded task and carries its evidence and correction route into the product decision.

01

Define

Promise

Task, source collection, language variety, system version and consequences outside scope.

02

Evaluate

Proof

Matched and local cases; missing evidence; prompt rules; visible failures and review disagreements.

03

Repair

Remedy

A usable escalation route, preserved evidence, accountable owner and dated re-evaluation.

This is a proposed operating contract, not a completed product evaluation or a legal standard. An untested slice remains unknown.

Kymata Labs framework informed by TyDi QA, Belebele, PAWS-X and MLQA evaluation guidance. [3] [4] [5] [6] [7]

Three questions for the same document

Return to the hypothetical municipal assistant, now as a release exercise. The source collection contains a dated list of acceptable documents and a notice saying some exceptions require staff review. The team is not testing whether the model can imitate a helpful official. It is testing three different commitments.

First, ask which documents the list names. An acceptable answer preserves the source's wording and conditions and identifies the source. A polished list with one invented addition fails. Second, ask whether an unlisted document will nevertheless be accepted. An acceptable answer acknowledges the boundary and gives the actual review route. Guessing from similar documents fails. Third, provide an older notice that conflicts with the current list. An acceptable answer follows the institution's explicit rule for dated sources or escalates the conflict if no rule resolves it. Choosing whichever passage is easiest to summarize fails.

Run equivalent intents in each tested language condition, but do not assume literal translations are equivalent questions. Reviewers should check that the request carries the same uncertainty and practical consequence. Add locally originated formulations that people would actually use. Keep a separate record of changes reviewers needed to make before the cases became comparable.

The exercise has no measured results here. Its value is that success can be defined without awarding points for a pleasant tone. A team can discover that one language condition needs retrieval work, another needs better handling of document names, and every condition needs a clearer escalation policy. Those are different repairs. A single label such as “multilingual quality” would make them harder to assign.

This is also where procurement can become useful. Ask a supplier to demonstrate the same commitments with the buyer's permitted source material and to preserve the review record. A generic benchmark may justify investigation. The service's own acceptance evidence should justify reliance.

Do not average away the person at the edge

A single overall score can be appropriate for research comparison while insufficient for a service decision. The TyDi QA documentation, for example, describes language-wise F1 scores averaged across languages excluding English. The aggregate has a specified purpose and denominator. It does not tell an individual user whether their particular task is supported. [3]

Product reporting should preserve the useful aggregate and the visible slices. Identify the tested language and variety, task family, sample size, review method and unresolved defects. Separate ordinary usefulness from consequential errors. A system can succeed at many routine requests while failing at the exception that changes a person's next action.

Do not report a clean percentage if the denominator excludes the cases the product cannot parse. Formatting failures, unanswered requests and escalations are part of the user experience. They can be classified separately, but they should remain visible. Conversely, a correct escalation should not automatically be scored as a useless answer when escalation is the intended behavior.

Small samples require restraint. Passing a short curated suite is evidence about those cases under those conditions. It is not proof of parity across an entire language. Review disagreement is also information: where qualified reviewers interpret the task differently, the product may need a clearer policy or a less decisive response rather than a more confident model.

The release process should preserve a dated record. Model changes, retrieval changes, source updates and revised translations can alter behavior without changing the language menu. MLQA's instructions to disclose evaluation settings and limit repeated test-set use are research practices with a practical analogue here: keep the acceptance set meaningful, and explain when tuning has made a reported result less independent. [6]

Where a service cannot support the full task, say what remains available. A message that explains a specific limit and offers a usable route forward is different from a vague disclaimer beneath a confident answer. A warning does not repair a wrong decision; a carefully scoped workflow may prevent the system from making it.

A service speaks through its remedies

The most provocative question is not whether AI can speak your language. Many systems can produce impressive multilingual output. The question is whether the institution behind the interface will accept an obligation in that language after the output matters.

A benchmark cannot answer that on the institution's behalf. It can test a defined capability, reveal exclusions and help locate weaknesses. Its contribution is diminished when a marketing claim turns a careful research boundary into a vague assurance of universal competence.

For buyers, the practical demand is modest in form and demanding in substance: show the task-specific evidence, the limits and the correction path together. For builders, the challenge is to make those three things evolve together. For users, the right question is less “How many languages do you speak?” than “What can I rely on you to do here, and who will help if you are wrong?”

A dropdown can open the door. It cannot stand behind the answer.

Evidence and method

This paper analyzes original benchmark repository documentation captured on November 16, 2026: XTREME, FLORES-200, TyDi QA, Belebele, PAWS-X and MLQA, plus Belebele's released prompting notes. We inspected the relevant task definitions, coverage, exclusions and reporting guidance. We did not read the full linked research papers, rerun their benchmarks, download and audit every dataset instance, test a current commercial model, interview users or measure a present-day language disparity.

The cited releases date principally from 2019–2024. FLORES-200's repository states that it is no longer updated; its coverage is described as that historical release, not the latest FLORES family. Repository commits and file hashes are preserved in the private evidence package. Dataset counts are descriptive properties reported by their authors, not new performance measurements.

The support contract, failure chain and accountability analysis are Kymata Labs proposals. The municipal-service example is invented to explain a mechanism and does not describe a real failure. No language, country or population is assigned a current reliability ranking. Search terms attached to this publication describe its subject; no search-volume or trending claim is made.

Questions worth asking.

Does this show that AI works poorly in every language except English?

No. We did not test current models or measure a present-day disparity. The paper examines what historical multilingual benchmark tasks establish and what they leave untested. English output also requires task-specific evidence and a usable correction path.

Why does the difference between TyDi QA and TyDiQA-GoldP matter?

The primary tasks allow no-answer outcomes. GoldP provides a passage guaranteed to contain the answer, removes unanswerable questions and excludes Thai and Japanese for tooling compatibility. Its results cannot independently establish no-answer behavior or performance in the excluded languages.

Can translating everything into English solve the problem?

It can be a useful design, but evaluate the full journey. Translation must preserve intent; retrieval must select relevant evidence; the answer must preserve limits; and the user must have a usable correction route. A fluent return translation does not verify source selection.

Should teams wait for perfect language parity before launch?

No universal parity prerequisite is proposed. Useful low-consequence functions can be released with specific limits, appropriate evidence and usable escalation. The release promise should match the task and its consequences; unsupported decisions can be restricted without removing every useful function.

What should a buyer ask a multilingual AI supplier to demonstrate?

Ask for the actual task, source collection, tested language varieties, system version, evaluation conditions and correction route. Review matched and locally originated cases, including missing or conflicting evidence. Keep sample sizes, failures and untested slices visible rather than relying on one language count or overall score.

SOURCES, NOT SCENERY

References.

1. **XTREME: task coverage and cross-lingual evaluation setting**
   Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat and Melvin Johnson · 2020 benchmark; repository accessed 2026-11-16 · Primary research repository documentation

   Original XTREME task coverage and evaluation setting. Most tasks transfer from English; BUCC and Tatoeba retrieval use pretrained representations directly. README and evaluate.py inspected. Not XTREME-R or a current model ranking.

   https://github.com/google-research/xtreme/blob/838c13b69daafb9328785d16caae2711e4012123/README.md

2. **FLORES-200: original machine-translation evaluation release**
   NLLB Team / Meta AI · 2022-06-30 release; repository accessed 2026-11-16 · Primary dataset release documentation

   Original FLORES-200 composition, splits and language-script inventory. The repository is no longer updated. Its 3,001 sentences and 842 source articles describe this historical release, not current FLORES+ coverage.

   https://github.com/facebookresearch/flores/blob/a6c830c6e1051fb4ac1a44b32358f00463f332bd/flores200/README.md

3. **TyDi QA: primary tasks, GoldP exclusions and evaluation guidance**
   Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev and Jennimaria Palomaki · 2020; GoldP v1.1 released 2020-02-27 · Primary research repository documentation

   Primary tasks permit no answer. GoldP removes unanswerable questions and excludes Thai and Japanese for tooling compatibility. README and changelog inspected; the authors require task-specific result names.

   https://github.com/google-research-datasets/tydiqa/blob/43cde6d598c1cf88c1a8b9ed32e89263ffb5e03b/README.md

4. **Belebele: parallel multiple-choice reading-comprehension task**
   Lucas Bandarkar and colleagues / Meta AI · ACL 2024 publication; repository accessed 2026-11-16 · Primary research repository documentation

   Parallel reading comprehension: 900 questions per variant, 122 variants and four options per question. Task composition, inventory and settings inspected. These are dataset properties, not product-reliability measurements.

   https://github.com/facebookresearch/belebele/blob/918890beb2290a8d3ef2d7a90369925959e1bacf/README.md

5. **PAWS-X: translated paraphrase data and split caveat**
   Yinfei Yang, Yuan Zhang, Chris Tar and Jason Baldridge / Google Research · EMNLP 2019; repository accessed 2026-11-16 · Primary dataset release documentation

   Human-translated evaluation pairs and machine-translated training pairs. The documented split caveat permits recurring first sentences but not identical complete pairs. Used to distinguish evaluation units.

   https://github.com/google-research-datasets/paws/blob/02b29f3af1143620d1b7f352247e65c0bdd2ec18/pawsx/README.md

6. **MLQA: question/context languages and evaluation disclosure**
   Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel and Holger Schwenk · 2019 research release; repository accessed 2026-11-16 · Primary research repository documentation

   Extractive QA across seven languages. Documentation distinguishes question/context languages, requires an answer-language setting and advises disclosure of evaluation conditions. Historical baseline scores are not used as current results.

   https://github.com/facebookresearch/MLQA/blob/24bf470b2e68f2bfb93b2fd31c370bb8ee18d9ed/README.md

7. **Belebele: zero-shot instruction and response-processing notes**
   Belebele research authors / Meta AI · Historical experiment notes; accessed 2026-11-16 · Primary evaluation protocol artifact

   Complete released instruction notes inspected. They document choice-letter prompts, model-specific formatting and response normalization. No resulting performance effect is estimated.

   https://github.com/facebookresearch/belebele/blob/918890beb2290a8d3ef2d7a90369925959e1bacf/sample_zero_shot_instructions.md