

Small Language Models for Less-Resourced Languages in a Real-World Scenario: The Case for Catalan

Roser Saurí[†], Josep Sànchez-Ferreres[★], Lluís Padró^{★†}, Josep Carmona^{★†}

[†] Process Talks S.L.

{roser, lluis, josep}@processtalks.com

[★] Computer Science Department

Universitat Politècnica de Catalunya - BarcelonaTech

{jsanchezf, padro, jcarmona}@cs.upc.edu

Abstract

Small Language Models (SLMs), typically ranging from a few million to 10–15 billion parameters, offer a promising solution towards constraints imposed by platform size—particularly mobile and IoT devices—and by the requirements of many organizations such as SMEs, which need solutions that ensure data privacy while remaining cost-effective. Their compactness and efficiency provide digital sovereignty and flexibility, though with more limited general-purpose capabilities. This makes them especially sensitive when working with underrepresented languages, such as Catalan, due to interference from majority languages that can increase bias risk. This paper evaluates state-of-the-art SLMs in a real-world Catalan use case: an AI assistant for older adults, assessing both user interactions and structured function call generation. Our work, which contributes to the [FREC](#) initiative for deploying a connected SLM-based infrastructure under the Model Context Protocol, demonstrates that some SLMs are able to deliver high-quality performance even in resource-constrained, linguistically minority environments.

Keywords: Small Language Models, model evaluation, Under-resourced languages

1. Introduction

The remarkable advances in AI could suggest that we are steadily approaching what is known as Artificial General Intelligence (AGI). In just a few years, model sizes have scaled to dimensions that would have been difficult to foresee until recently, supported by the availability of vast bodies of knowledge and the computational capabilities to process them efficiently, which has resulted into the so-called Large Language Models (LLMs).

However, there are clear signs indicating that AGI will not be achieved through the development of ever larger, increasingly multi-capable models trained incrementally on huge quantities of data, but rather through the opposite approach ([AAAI Presidential Panel, 2025](#); [Rashid and Rehman, 2025](#); [Brodsky, 2025](#)). Two main reasons supporting that are particularly relevant here: firstly, the need for a network of specialized components; secondly, the need for small language models.

Need for a network of specialized components.

Centralizing all knowledge and cognitive skills in a single universal system is not feasible. Rather, a structure of interconnected components operating in continuous cooperation appears to be more realistic. Several situations require this, for instance:

- *Accessing highly specialized or organization-specific knowledge*, e.g., the log of incidents in the assembly process of an automotive company's maintenance service, in order to identify

appropriate actions in similar situations.

- *Accessing periodically updated knowledge*. For example, applications that publish information about weather conditions, stock market fluctuations or product reviews.
- *Integrating AI with corporate applications*, such as a restaurant booking calendar or a company's billing management system.

Need for small models. A second reason why an AGI based on massive models is unrealistic lies in the size constraints imposed at different levels; primarily, the limited dimensions of many platforms on which AI must be hosted, such as mobile and IoT (Internet of Things) devices, as well as the technological constraints faced by many SMEs, which recognize the potential of AI to optimize management or enhance their value proposition, but must ensure absolute data privacy and often cannot afford the infrastructure required to run LLMs locally.

All these factors call for the use of lightweight models, known as Small Language Models (SLMs). Typically ranging from a few million parameters up to around 10-15 billions, SLMs are designed to be more compact and efficient than large-scale models, both in terms of parameter count and computational requirements. Thus, they provide flexibility and an optimal level of digital sovereignty, as needed by many organizations. However, their smaller size means they exhibit lower cognitive ca-

capacity and narrower general-purpose scope, and so cannot adapt to user requirements with the same level of autonomy as LLMs (Corradini et al., 2025).

This is particularly problematic when working with minority and underrepresented languages. The clear disparity in the availability of digital data across languages results in interferences from majority languages on the generation of text for minority ones, a bias that gets amplified with less able models like most SLMs.

The aim of this paper is to assess whether state-of-the-art SLMs are suitable for real-world applications involving minoritized languages, with a particular focus on Catalan. To this end, we evaluate the performance of several models, both commercial and open source, deployed at the core of an assistant designed for older adults. We chose this use case because it allowed us to clearly delimit the experimental scope through well-defined intents (e.g., send a message, set a reminder), while simultaneously testing two key capabilities: (a) the ability to interact with users and provide appropriate responses; and (b) the ability to function calling—that is, identifying required follow-up actions and representing them through structured outputs.

This research forms part of a broader project, called FREC¹, aimed at deploying an infrastructure of specialized and interconnected components for Catalan based on the Model Context Protocol (MCP).² Its purpose is to develop a platform of connected AI assistants, primarily based on SLMs, in order to address common platform-size and the economic constraints faced by many organizations, while supporting the use of Catalan in digital applications. Its ultimate goal is to equip the social and economic fabric of the Catalan-speaking community with technological resources that help streamline and optimize business processes and, more broadly, knowledge-related activities currently available mainly in English.

2. Related Work

Our research addresses two areas within AI: (1) the use of language models in resource-constrained environments, and (2) solutions to expand the presence of minoritized languages, such as Catalan, in language models.

SLMs in small environments. There are currently several SLMs designed to offer an optimal balance between efficiency, speed, and capability on the one hand, and compact size on the other. Even if developed by large corporations, most of them are open source, such as: Mistral 7B (Mistral AI), Llama

3 8B (Meta), Salamandra (Barcelona Supercomputing Center), Gemma (Google), Qwen2.5 and Qwen3 (Alibaba), Phi-3 (Microsoft), etc. Among all those, only Gemma 3 1B (Ikonomidis et al., 2025), Phi-3 mini (Bilenko, 2024), and Qwen2.5³ were specifically designed to operate on small platforms.

There is growing interest in that type of models, which is reflected in ongoing efforts to design new, better-optimized ones (Shakhadri et al., 2025) or to obtain them through compression of larger models (Wang et al., 2025; Xu et al., 2025), as well as in studies evaluating their performance, both in terms of capabilities and efficiency, within resource-constrained environments (Murthy et al., 2024, Laskaridis et al., 2024).

Our exploration is more closely aligned, however, with the work of Lu et al. (2025), which evaluates a large number of SLMs based on benchmark datasets—with the key difference here that our study focuses on a specific use case and for a language other than English.

SLMs and minoritized languages. In parallel to all that work, there are also a number of initiatives towards making language models more capable cross-linguistically, be it either by means of different kinds of adaptation techniques (Gurgurov et al., 2025; Shang et al., 2025) or by building new models with a specific linguistic design. In the current case, it is worth mentioning the Salamandra family,⁴ a set of open-source multilingual models, available in 2B and 7B versions, developed at the Barcelona Supercomputing Center within the framework of the Aina Project, dedicated to *"boosting the Catalan language in the digital era."*⁵ They are models trained for European languages, with a particular focus on Catalan and the other co-official languages of Spain (alphabetically, Basque, Galician, and Spanish).

A broad overview of all the activity around language models for under-resourced and minoritized languages can be found in the proceedings of the First Workshop on Language Models for Low-Resource Languages (LoResLM) (Hettiarachchi et al., 2025). Our work builds on previous investigations on language model performance for Catalan by the same authors (Padró and Saurí, 2024), where the focus was on fine-tuning LLMs to produce high-precision semantic representations.

¹<https://frec.cat>

²<https://www.anthropic.com/news/model-context-protocol>, <https://github.com/modelcontextprotocol>

³<https://github.com/QwenLM/Qwen2.5-Omni>

⁴<https://langtech-bsc.gitbook.io/ainakit/models/models-de-text>

⁵<https://projecteaina.cat/en/>

3. Experimental Design

3.1. Use Case: An AI Assistant for Older Adults

The goal of this research was to assess whether state-of-the-art SLMs are suitable for real-world applications involving minoritized languages, with a particular focus on Catalan. To evaluate the performance of different SLMs, we deployed them at the core of an AI assistant for older adults called Bleta, developed by Bleta Solutions.⁶

This use case is of particular interest to us due to its societal impact. In today's world, people are increasingly living in isolation or in socially fragmented contexts, which particularly affects older adults. In this context, technology can serve as a means of connection and communication, as well as a gateway to many other resources. Unfortunately, a significant proportion of older adults have limited familiarity with digital technologies, which exacerbates the digital divide. In this respect, AI-based assistants such as Bleta can help mitigate this limitation by enabling more natural and accessible forms of interaction, particularly when communicating in the user's own language.

For our experimental purposes, this use case allows us to assess two key capabilities in SLMs:

- The ability to interact with users and provide appropriate responses; and
- The ability to identify required follow-up actions and represent them through function call structured outputs.

We delimited the experiment to 5 intents:

- Sending a message (`send_message`)
- Making a phone call (`make_call`)
- Getting a weather forecast (`get_weather`)
- Setting a reminder (`set_alarm`)
- Asking for directions (`navigate_to`)

Next is their representation schema (simplified for space constraints' sake):

```
1 [{"function": {
2   "name": "send_message",
3   "parameters": ["name", "number",
4                 "message"]},
5 {"function": {
6   "name": "make_call",
7   "parameters": ["name", "number"]},
8 {"function": {
9   "name": "get_weather",
10  "parameters": ["location",
11               "timeframe"]},
12 {"function": {
13  "name": "set_alarm",
14  "parameters": ["message", "day",
15               "time"]},
```

⁶<https://bleta.io/>.

```
{ "function": {
  "name": "navigate_to",
  "parameters": ["location"]}}
```

These intents involve different degrees of complexity depending on the parameters needed. For instance, `get_weather` involves only 1 parameter while `send_message` implies 3.

3.2. Test Dataset

The test dataset was fully crafted by hand. It includes small **dialogues** for each intent, which consist of several **exchanges**, i.e., pairs of user-assistant **turns**. The very first exchange contains the **user request** (e.g., *I'd like to send a message to Maria*). If this request already provides all the information needed to instantiate the corresponding function call, the AI assistant returns that call and the interaction concludes.⁷ This represents the simplest form of dialogue, as illustrated below:⁸

Assistant: Good morning, this is your assistant. How can I help you today?
User: What's the weather like in Vic?
A: Let me find it.
[function call] `get_weather(vic)`

Example 1: Intent `get_weather`

By contrast, if the original user request does not provide all the necessary parameter information, the assistant must request it, leading to additional user–assistant exchanges. In the next example, the value for parameter `location` is initially missing:

U: Can you help me get to a place?
A: Sure! Where would you like to go?
U: To the closest taxi stop.
A: No problem. Here you have!
[fc] `navigate_to(taxi_stop)`

Example 2: Intent `navigate_to`

Other dialogues are even more complex:

⁷There is indeed a final required step, which consists in retrieving the output of that function call and generating the response to the user. We do not include this step here, as it involves the execution of independent applications.

⁸For simplification's sake, these and the following dialogue examples are reproduced in English although the original data are in Catalan.

```

U: I'd like to send a message to Maria.
A: Let me find her telephone number.
      [fc] get_contacts (Maria)
Tool: [Providing user's contact list]
A: There are 2 contacts called Maria
      in your contacts list: Maria Pi and
      Maria Bartra. Which one would you
      like to message?
U: Maria Pi
A: What is your message for her?
U: Ask her if she'll come tomorrow.
A: Ok, doing it right now.
      [fc] send_message ("Maria Pi",
        655332925, "Maria, will
        you come tomorrow?")

```

Example 3: Intent `send_message`

In the dialogue above we see that, since users are not expected to know the addressee's telephone number, the assistant will need to activate a further function call (`get_contacts`) to obtain the contacts list from which to fetch it. Note also that there are additional follow-up exchanges with the user, first in order to disambiguate the addressee's name (*Which Maria would you like to message?*) and later to ask for the message content given that the user didn't provide it initially.

When creating the test dataset, efforts were made to ensure that each intent included examples of varying lengths and levels of complexity, ranging from trivial cases (like Example 1) to highly under-specified or ambiguous ones (Example 3). That naturally resulted in dialogues of different length.

Table 1 presents the test dataset from a quantitative perspective. That is, for each intent, number of dialogues (#D), number of exchanges (#EX; i.e., user-assistant pairs), dialogue minimum and maximum length (assessed in terms of number of exchanges per dialogue: EX/D), and total number of function calls expected to be returned by the AI assistant (#FC).

intent	#D	#EX	EX/D		#FC
			min	max	
message	12	41	2	5	24
call	8	21	1	4	13
weather	13	20	1	2	13
alarm	16	34	2	3	16
navigate	6	9	1	2	6
Average	11.0	25.0	1.4	3.2	14.4
Total	55	125	-	-	72

Table 1: Test dataset figures (#D: dialogues; #EX: exchanges; EX/D: number of exchanges per dialogue; #FC: function calls.

As can be seen, intents `send_message` and `make_call` are more complex than the other three: they have a larger maximum number of exchanges

and, more importantly, may involve more than 1 function call per dialogue.

4. Setup

4.1. AI Assistant Deployment

The experiments were conducted using an AI assistant mock-up that emulates the five intents above as implemented in the real application Bleta. Although this is a simulated setting, the testing framework was designed to mimic the real LLM orchestration engine behind the FREC project, which powers the Bleta application. Thus, the conclusions drawn from the evaluation remain applicable to the actual production environment targeted by our broader project, since the operational capacities of the SLMs are identical in both contexts.

To ensure proper interaction with MCP servers, the model was prompted to generate a JSON document specifying the MCP server and tool to call, along with its arguments. The JSON document had to be enclosed within a pair of special markers (`<tool_call>` and `</tool_call>`) to allow the system to extract the tool call from the LLM's textual output.

This mock-up is a prompt-based assistant implemented to mimic the behavior of OpenAI tool calling functionality, a mechanism that allows models to generate structured requests to predefined external functions, enabling integration with software systems and real-world actions.⁹

The assistant prompt was engineered to ensure maximum precision. It includes instructions on:

- Assistant's profile: Dedicated to support older adults and make their life easier.
- Assistant's function: Chat with users or support them on certain tasks. In the latter case, the assistant is never expected to execute the action but rather find the adequate function call for an external tool to execute it, where all parameters must be instantiated with proper data. The assistant is also provided with the list of possible functions.
- Intents: Each of the 5 selected intents has its own well-delimited block of instructions. In addition, there are instructions for a few other intents that are not evaluated here: remember user's relevant data, search for an app, bring an app to the main screen.¹⁰

⁹<https://platform.openai.com/docs/guides/function-calling>

¹⁰The last 2 intents are motivated by the fact that the AI assistant is expected to operate within a tablet specifically designed for older adults.

- Other cross-intent instructions are also provided, such as: how to get a contact telephone number (relevant for intents `send_message` and `make_call`) or how to format dates and time expressions (applicable to `get_weather` and `set_alarm`).
- Language: Interaction with users must be in Catalan. The prompt however is written in English because in previous experiments we could see that this resulted in better performance than if expressed in Catalan.
- General behavior: Commands such as be polite and respectful, be concise, do not invent, do not speak technically, and the like.
- Output format: The function call must be returned embedded in tags for easy retrieval and structured as a JSON object.

The prompt takes 6232 tokens: 4309 are in the instructions part and 1923 in the functions listing.¹¹

4.2. Experimental Settings

Evaluated SLMs are deployed on a dedicated machine with Intel® Core™ i5-13500 CPU, 64GB of DDR4 RAM and a NVIDIA RTX™ 4000 SFF Ada Generation GPU with 20GB of GDDR6 VRAM, and loaded either directly with HuggingFace transformers library (16-bit weights), or via Ollama (quantized to 8 or 4 bits). The cost of such machine in a cloud provider is of 184€ plus VAT per month.

Only one model is loaded into the GPU at a given time, so experiments were run sequentially, unloading each used model before loading the next one.

Both with transformers or Ollama, a chat-style inference method is used to feed the model with a prompt containing the full conversation history with user and system messages, including called tool results and model answers. All models tested where run with the same parameters: `temperature 0.1`, number of context tokens `num_ctx=16384`, maximum tokens to generate `maxtok=2048`.

4.3. Models Selected

Our aim was to conduct a comprehensive evaluation; accordingly, we tested a large number of SLMs. The selection criteria were as follows:

1. Set an upper boundary of 14B parameters for model size; models exceeding this threshold were not considered SLMs.
2. Within that size limit, select the largest model from each model family.

3. If the tested model performs reasonably well (either precision or recall above 60%), evaluate a smaller model from the same family when available.
4. Continue testing progressively smaller models within the same family until performance falls below the acceptance threshold.

Moreover, we ran the same experiments using several top-tier commercial LLMs, namely OpenAI's `gpt-4o`, `gpt-5`, `gpt-5-mini`, and `gpt-5-nano`. Although these are not SLMs, they served as a reference ceiling for performance.

Overall, we tested 40 models among the following families: `cogito`, `deepseek`, `dolphin`, `gemma`, `glm4`, `gpt-oss`, `llama`, `minis-tral`, `mistral`, `nexusraven`, `olmo`, `phi`, `qwen`, `salamandra`.

5. Evaluation

5.1. Evaluation Methodology

We evaluated the performance of multiple SLMs when deployed at the core of the AI assistant described in Section 4.1. Upon receiving the initial user request, the assistant was required to either:

- (a) return a fully populated function call (as in Example 1), the correction of which was then assessed; or,
- (b) if information was missing, request the necessary details from the user (Examples 2 and 3). User responses to those requests were then automatically supplied to the assistant from the test dataset so that eventually the adequate function call could be fully populated.

A key challenge in automatically assessing the assistant's capability here is that, when multiple pieces of data were missing from the original request, the order in which the assistant asks the necessary questions may differ from the order used in the test set. Take for instance the dialogue for sending a message in Example 3: both the addressee and the message content are absent in the initial user request. The dialogue in the test dataset follows a fixed sequence (first requesting the addressee and then the message content), but the assistant could reasonably reverse this order.

Following a different order does not mean an incorrect result as long as the questions are adequate to the user's initial request. Therefore, we needed to ensure that the evaluation approach did not punish that. To that end, we applied the following evaluation process: whenever multiple dialogue exchanges were needed, each new input to the assistant would accumulate the prior context from the

¹¹According to <https://platform.openai.com/tokenizer>.

test set, and not the context generated at the previous user-assistant exchanges. That allowed us to impose the sequence order from the ground truth and therefore guide the assistant’s reaction.

5.2. Evaluated Capabilities

We evaluated the SLMs ability for both:

1. Properly interacting with users; that is, asking for clarification or additional information,
2. Returning the correct function call.

In addition, we also run a qualitative assessment on the SLMs linguistic competence in Catalan. We address these aspects next.

5.2.1. Returning the correct function call

Model performance is evaluated by comparing the output JSON structure with the ground truth, and computing precision (P), recall (R), and F1 scores for each intent as well as overall. For the comparison’s sake, both structures are flattened and minimally normalized (lower-cased, some punctuation removed, etc.).

Moreover, assessing the correctness of text fields (such as attribute `message` in intents `send_message` and `set_alarm`) was more involved than just a plain match, given that the same content can be expressed in different yet equally valid ways. Those fields were therefore evaluated using the LLM-As-A-Judge methodology (Li et al., 2025; Gu et al., 2026), whereby an LLM is prompted with guidelines to act as the evaluator.

To that end, we used `gpt-4o` prompted to identify whether two texts can be accepted as paraphrases of the same content, distinguishing among three possible situations: `EXACT` match, `PARTIAL` match, and `MISMATCH`. Both `EXACT` and `PARTIAL` matches were considered correct, as we observed that most partial deviations from the ground truth were minor and did not affect the intended meaning.

5.2.2. Proper interaction with users

Throughout the user-assistant conversation, the assistant’s utterances can be of two types: either requests for additional information or feedback to the user (e.g., *Sure, let me find it!* or *I’m sorry but I’m unable to help with this request.*). The appropriateness of these responses is just as important as the quality of the generated function calls, given the assistant’s role as a support tool—particularly in this case, for older adult users.

We therefore evaluated precision (P), recall (R), and F1 for the two utterance classes—`REQUEST` and `FEEDBACK`—both by intent and overall. Additionally, for utterances that matched the class in the test set, we assessed whether their content was equivalent

using the LLM-As-A-Judge methodology. As before, content matching was categorized as `EXACT`, `PARTIAL`, or `MISMATCH`.

6. Results

Model	Par	Quan	mP	mR	mF1	Time
cogito	8b		40.5	59.7	48.3	3.5
cogito	14b		62.9	41.0	49.7	5.0
deepseek-llm	7b		42.4	19.4	26.6	7.6
deepseek-r1	14b		42.4	19.4	26.6	35.1
dolphin3	8b		41.9	45.8	43.8	88.8
gemma	7b		32.3	26.4	29.0	3.8
gemma2	2b		33.7	31.5	32.6	2.1
gemma2-it	9b	q4	63.3	65.6	64.4	4.5
gemma3	4b		41.3	58.2	48.3	5.3
gemma3	12b		64.1	45.1	52.9	13.0
gemma3n	4b		37.5	47.6	41.9	38.6
glm4	9b		47.4	23.1	31.0	3.3
gpt-oss	20b		74.3	54.9	63.2	8.1
llama3	8b		32.9	42.9	37.2	4.5
llama3.1	8b		42.0	30.8	35.5	5.4
llama3.2	3b		35.8	31.5	33.5	2.3
ministral-3	8b		64.2	78.0	70.4	3.9
ministral-3-it†	14b	q4	85.1	81.7	83.4	6.4
ministral-3-it‡	14b	q8	70.2	85.3	77.0	7.8
mistral	7b		22.0	23.8	22.8	13.2
nexusraven	13b		38.6	20.5	26.8	88.8
olmo2	13b		9.8	16.5	14.0	21.1
olmo-3	7b		51.1	32.6	39.8	140.6
phi3	14b		43.2	20.9	28.1	5.7
phi3.5	3.8b		26.5	19.0	22.2	10.8
phi4	14b	q4	71.0	58.2	64.0	8.2
phi4-mini	3.8b		40.3	36.6	38.4	2.7
phi4-mini-rs	3.8b		42.1	19.4	26.6	41.7
*phi4-mini-it	4b		51.3	37.0	43.0	8.3
phi4-rs	14b		34.6	34.1	34.3	10.3
qwen2.5-it	7b		50.2	49.8	50.0	3.5
qwen2.5-it	14b		78.7	67.8	72.8	7.2
qwen3	4b	q4	76.4	80.6	78.4	23.9
qwen3	4b	q8	77.0	64.8	70.4	31.9
qwen3-th‡	4b	q4	84.2	72.2	77.7	42.5
qwen3	8b	q4	84.6	86.4	85.5	28.1
qwen3	8b	q8	87.2	90.1	88.6	38.7
qwen3	14b	q4	88.9	91.2	90.1	43.8
qwen3	14b	q8	84.3	90.5	87.3	58.2
*salamandra-it§	7b		49.0	52.4	50.6	10.0

Table 2: Screening phase. Average metrics and runtime (minutes) across all intents, on the skill of function call generation. **Legend:** (a) Model name prefix *: run via HF transformers. (b) Model name suffixes: `-it`: instruct model; `-rs`: reasoning model; `-th`: thinking model. (c) Version symbols: †: v.2512; ‡: v.2507; §: v.20260108.

Experiments were conducted in two phases. In the first phase, which we will refer to as the **screening phase**, each model was tested once on its ability to generate function calls. Models achieving an F1 score above 75% were then subjected to the

10-run experiment phase in order to obtain average performance metrics, both for function calls accuracy and for interaction quality with the user.

Additionally, we ran the same experiments on top-performing commercial models to establish a reference ceiling for our results.

6.1. Screening Phase

The results of this phase are presented in Table 2. They report the average of micro-precision (mP), micro-recall (mR) and micro-F1 (mF1) across all intents in the test dataset, as well as the time (in minutes) required to process the entire dataset, which consists in 125 dialogue exchanges.

Models are identified by their name, number of parameters (Par) and quantized form (Quan). When relevant, the name is followed by a suffix indicating its variant (*-it* for *instruct*, *-rs* for *reasoning* and *-th* for *thinking*), as well as a symbol denoting the version date. Finally, models executed via Huggingface transformers are marked with a prefixed \star .

The best results (scores above 75%) are shown in bold. These are concentrated across different variants of two model families: `qwen3` and `ministral-3`. Within the former, high-performing models span multiple sizes (4B, 8B, and 14B) whereas for the latter, only the 14B variants reach the top tier, and they do not attain the same level of performance as the strongest `qwen3` models. In contrast, the best-performing `ministral-3` models demonstrate greater efficiency in terms of runtime, requiring between 6 and 8 minutes to process the test dataset, compared to a range of 28 to over 58 minutes for the top `qwen3` models.

Finally, underlined results indicate models that were unable to generate any function calls. Results falling below that threshold appear in italics. They correspond to models that attempted to generate function calls but largely failed to do so correctly.

As a **reference ceiling**, Table 3 presents the corresponding results obtained with top state-of-the-art commercial models. The best-performing models here are `gpt-5-mini`, followed by `gpt-5.2`. Considering however the substantial difference in runtime (26.9 minutes vs. 4.1), the latter emerges as the clear winner in terms of efficiency within its model group.

Model	mP	mR	mF1	Time
gpt-4o	88.8	72.5	79.8	3.1
gpt-5-mini	93.0	92.7	92.8	26.9
gpt-5-nano	87.9	87.5	87.7	77.8
gpt-5.2	93.4	88.6	91.0	4.1

Table 3: Performance metrics and runtime (min.) for some GPT models on function call generation.

The comparison between the SLMs in Table 2 and the commercial LLMs in Table 3 is of great interest. Despite the significant difference in model size, the top-performing SLMs (in particular within the `qwen3` family) achieve scores comparable to—or sometimes even exceeding—those of certain LLMs. In this case, however, runtime scores are always substantially worse.

6.2. 10-Run Experiment Phase

Building on the results above, we selected the best-performing SLMs and ran a series of 10 iterations for each. In this second phase, we evaluated not only their performance in function call generation, but also their ability to interact effectively with the user and their linguistic competence in Catalan.

6.2.1. Function Call Generation

Table 4 presents the SLMs performance results on function call generation.

Model	Par	Quan	mP	mR	mF1	Time
<code>ministral-3-it†</code>	14b	q4	89.5	82.1	85.6	4.7
<code>ministral-3-it†</code>	14b	q8	78.0	83.0	80.4	7.2
<code>qwen3</code>	4b	q4	76.6	78.8	77.7	23.4
<code>qwen3-th†</code>	4b	q4	78.2	79.7	78.9	24.1
<code>qwen3</code>	8b	q4	83.2	86.4	84.8	25.7
<code>qwen3</code>	8b	q8	89.5	89.5	89.5	39.2
<code>qwen3</code>	14b	q4	85.8	91.6	88.6	45.1
<code>qwen3</code>	14b	q8	83.7	90.5	87.0	57.4

Table 4: Results on function call generation from the 10-run experiments. Same legend as Table 2

Both the best- and worst-performing SLMs belong to the `qwen3` family (`qwen3 8B` with q8 quantization and `qwen3 4B q4`, respectively). Notably, the 8B model outperforms larger variants on this task—even within the same family.

The `ministral-3-instruct 14B q4` model also delivers solid performance, comparable to `qwen3 8B q4`, while surpassing it in terms of time efficiency, a critical matter for our use case.

6.2.2. Ability to Interact with the User

A second key aspect to assess in the context of our use case is SLMs' capability to interact appropriately with users. Two main types of interactions are considered: (a) requesting missing information needed to complete the function call (tagged as `REQUEST`) and (b) providing feedback to the user regarding the current task (`FEEDBACK`). Table 5 presents the results in terms of generating appropriate utterances in response to user input. The winning model on this performance aspect is again `qwen3 8B` with q8 quantization.

			Class REQUEST			Class FEEDBACK			Content Matching			
model	par.	quan.	mP	mR	mF1	mP	mR	mF1	FULL	PARTIAL	FULL+PART	MISMATCH
ministral-3-it†	14b	q4	83.3	94.9	88.7	100.0	6.6	12.4	39.9	47.5	87.4	12.6
ministral-3-it†	14b	q8	85.5	71.2	77.7	100.0	1.4	2.8	39.1	55.2	94.3	5.7
qwen3	4b	q4	96.7	72.1	82.5	0.0	0.0	0.0	43.9	39.4	83.3	16.7
qwen3-th‡	4b	q4	97.7	71.6	82.5	0.0	0.0	0.0	44.0	40.4	84.4	15.6
qwen3	8b	q4	97.8	84.7	90.7	90.0	1.4	2.8	38.7	51.7	90.3	9.7
qwen3	8b	q8	98.9	87.9	93.1	100.0	1.5	3.0	48.7	44.2	92.9	7.1
qwen3	14b	q4	98.4	75.6	85.4	100.0	1.4	2.8	50.5	43.7	94.1	5.9
qwen3	14b	q8	97.9	79.0	87.4	100.0	1.4	2.8	58.7	36.9	95.6	4.4

Table 5: Results from the 10-run experiments. Performance on the ability to interact with the user. Same legend as in Table 4

Overall, the situation is highly uneven. While all SLMs perform very well at identifying when a request is needed, the opposite is true for feedback-type interactions, resulting in extremely low recall scores across the board. We believe this is due to the way the prompt was constructed: considerable emphasis was placed on enabling the SLM to detect when additional information was required (and thus to produce request-type utterances), whereas little attention was given to instructing the model to provide feedback to the user at each step.

It is, however, noteworthy that `ministral-3-instruct 14B q4` showed a somewhat greater natural ability to recognize the need for this type of user interaction. It is also the model that reaches the best recall score for class REQUEST.

Finally, we evaluated the percentage of content match for the utterances in cases of correct class (right-hand side part of Table 5). Although the classification here is three-fold (full match, partial, and mismatch), we decided to group together full and partial matches because in most cases partial matches are also legitimate utterances that appropriately address the user’s conversational needs. Here, the best result was achieved by the `qwen 14B` with q8 quantization.

6.3. Linguistic Competence

Throughout this work we assumed that the SLMs’ ability both to understand users requests and to interact appropriately could serve as a proxy for assessing their linguistic competence in Catalan—at least within the specific context defined by the use case under consideration.

However, is that truly accurate? The fact that the conversation may have flowed more or less successfully is not necessarily indicative of the correctness of the generated text (i.e., free from spelling or grammatical errors, or interference from other languages). By contrast, linguistic quality is essential to ensure effective communication at all times.

To determine the extent to which there is a correlation between SLMs’ quantitative performance and

their linguistic competence, a Catalan philologist assessed the quality of the generated utterances by the winning models in each family (`qwen3 8B q8` and `ministral-3 14B 4q`), looking at the different types of errors: spelling, grammatical, pragmatic coherence, lexical choice, and finally other language’s interferences.

It was noticed that, although in quantitative terms the `ministral-3` model exhibits a higher number of linguistic errors than the `qwen3` model, the degree of severity—and therefore the potential to negatively affect communication with the user—is greater in the latter. Many of the errors in `ministral-3` are spelling in nature and have limited impact in a voice assistant setting. And although this model also shows a non-negligible number of grammatical errors (e.g., related to verb subcategorization or preposition choice), which may affect communication more substantially, in real usage some of these cases may be masked due to phonological processes (e.g., vowel elision). Moreover, in several instances, the errors are recurrent and could therefore be mitigated through additional instruction of the model.

By contrast, the errors observed in `qwen3` are more detrimental, as many are pragmatic in nature and thus affect the coherence of the utterance content. For example, incorrect uses of grammatical person (e.g., “*I’d like to call Maria Pi?*” instead of “*Would you like to call Maria Pi?*”). The model also displays a higher degree of interference from Spanish, including omission of clitic pronouns, omission of definite articles, calques, and the direct use of Spanish lexical items. These issues not only lead to potential confusion in meaning, but also reduce the naturalness of conversations and, more importantly, may undermine the user’s trust in the assistant.

7. Conclusion and Future Work

This study examined the suitability of state-of-the-art SLMs for real-world applications in minoritized languages, with a specific focus on Catalan. To this end, we analyzed the performance of several

models integrated into the core of an AI assistant designed for older adults.

We have found that models from at least two families are capable of responding very positively to this challenge, reaching levels (almost) comparable to top-tier commercial LLMs: `qwen3` and `ministral-3`; in particular, `qwen3 8B` quantized to q8, and `ministral-3 14B` quantized to q4.

Regarding the ability of function call generation, the former excelled in overall performance while the latter, which achieved slightly lower marks, demonstrated a remarkable level of efficiency.

As for the ability to interact with the user, the results revealed a clear deficiency in the models' instructional prompt. Although it correctly guides the SLMs to request missing information when needed, it lacks an adequate level of guidance about providing feedback to the user—a shortcoming that we plan to address in the near future.

A further interesting insight from this research emerged from the qualitative analysis of the winning models' linguistic competence, in comparison with the quantitative results. While the latter indicate a certain superiority of the `qwen3 8B` model over `ministral-3 14B`, the qualitative evaluation conducted by a Catalan philologist determined that the types of linguistic errors produced by the former are more detrimental than those of the latter in terms of maintaining fluent communication with the human user and—crucially for our use case—earning and preserving the user's trust. These conclusions are, however, preliminary and will be the subject of further research.

8. Acknowledgments

This piece of work has been supported by MCIN/AEI/10.13039/501100011033 under grant PID2020-112581GB-C21 (MOTION). Also, by the European Union – NextGenerationEU, Spain's *Recovery, Transformation and Resilience Plan* (RRP), in collaboration with the Catalan government (Generalitat de Catalunya), under grant SDC007/25/000055.

We are also very thankful to Bleta Solutions for accepting to collaborate with us, and for their willingness to share part of their working data.

9. Bibliographical References

- AAAI Presidential Panel. 2025. [Future of AI research](#). Technical report, Association for the Advancement of Artificial Intelligence (AAAI). Accessed: 2026-02-19.
- Misha Bilenko. 2024. [Introducing Phi-3: Redefining what's possible with SLMs](#). Accessed: 2026-02-19.
- Sascha Brodsky. 2025. [Beyond big models: Why AI needs more than just scale to reach AGI](#). *IBM Think News*.
- Flavio Corradini, Matteo Leonesi, and Marco Piangerelli. 2025. [State of the art and future directions of small language models: A systematic review](#). *Big Data and Cognitive Computing*, 9(7).
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. 2026. [A survey on LLM-as-a-Judge](#). *The Innovation*.
- Daniil Gurgurov, Ivan Vykopal, Josef van Genabith, and Simon Ostermann. 2025. [Small models, big impact: Efficient corpus and graph-based adaptation of small multilingual language models for low-resource languages](#). *arXiv preprint arXiv:2502.10140*.
- Hansi Hettiarachchi, Tharindu Ranasinghe, Paul Rayson, Ruslan Mitkov, Mohamed Gaber, Damith Premasiri, Fiona Anting Tan, and Lasitha Uyangodage, editors. 2025. [Proceedings of the First Workshop on Language Models for Low-Resource Languages](#). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates.
- Marissa Ikonmidis, T.J. Alumbaugh, Mark Sherwood, and Cormac Brick. 2025. [Gemma 3 on mobile and web with Google AI Edge](#). <https://developers.googleblog.com/en/gemma-3-on-mobile-and-web-with-google-ai-edge>. Accessed: 2026-02-19.
- Stefanos Laskaridis, Kleomenis Katevas, Lorenzo Minto, and Hamed Haddadi. 2024. [MELTing point: Mobile evaluation of language transformers](#). *arXiv preprint arXiv:2403.12844*.
- Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. [From generation to judgment: Opportunities and challenges of LLM-as-a-judge](#). *arXiv preprint arXiv:2411.16594*.
- Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Wei Liu, Jian Luan, Xiwen Zhang, Nicholas D. Lane, and Mengwei Xu. 2025. [Demystifying small language models for edge deployment](#). In *Proceedings of the 63rd Annual Meeting of the ACL*. ACL.

- Rithesh Murthy, Liangwei Yang, Juntao Tan, Tulika Manoj Awalgaonkar, Yilun Zhou, Shelby Heinecke, Sachin Desai, Jason Wu, Ran Xu, Sarah Tan, Jianguo Zhang, Zhiwei Liu, Shirley Kokane, Zuxin Liu, Ming Zhu, Huan Wang, Caiming Xiong, and Silvio Savarese. 2024. [MobileAIBench: Benchmarking LLMs and LMMs for on-device use cases](#). *arXiv preprint arXiv:2406.10290*.
- Lluís Padró and Roser Saurí. 2024. [Fine-tuning open access LLMs for high-precision NLU in goal-driven dialog systems](#). In *Proceedings of the Second International Workshop Towards Digital Language Equality (TDLE): Focusing on Sustainability @ LREC-COLING 2024*, pages 33–42, Torino, Italia. ELRA and ICCL.
- Mehmood Rashid and Eid Rehman. 2025. [The Illusion of Intelligence: Evaluating Large Language Models Against Grounded Criteria of Artificial General Intelligence](#). *Preprints*. Publisher: Preprints.
- Syed Abdul Gaffar Shakhadri, Kruthika KR, and Rakshit Aralimatti. 2025. [SHAKTI: A 2.5 billion parameter small language model optimized for edge AI and low-resource environments](#). *arXiv preprint arXiv:2410.11331*.
- Guokan Shang, Hadi Abdine, Yousef Khoubrane, Amr Mohamed, Yassine Abbahaddou, Sofiane Ennadir, Imane Momayiz, Xuguang Ren, Eric Moulines, Preslav Nakov, Michalis Vazirgiannis, and Eric Xing. 2025. [Atlas-Chat: Adapting large language models for low-resource Moroccan Arabic dialect](#). In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 9–30, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jiacheng Wang, Yejun Zeng, Jinyang Guo, Yuqing Ma, Aishan Liu, and Xianglong Liu. 2025. [SLMQuant: Benchmarking small language model quantization for practical deployment](#). In *Proceedings of the 3rd International Workshop on Rich Media With Generative AI*, page 2–10. ACM.
- Mingxue Xu, Yao Lei Xu, and Danilo P. Mandic. 2025. [TensorSLM: Energy-efficient embedding compression of sub-billion parameter language models on low-end devices](#). *arXiv preprint arXiv:2506.13514*.