Small language models explained: when smaller beats bigger

26 October 2026 · 5 min read · blogs.iictp.org/small-language-models-explained

A small language model (SLM) is a language model small enough to run on modest hardware, such as one workstation, a phone or an edge device. There is no agreed size cutoff; people usually mean models from a few hundred million up to a handful of billions of parameters, against the much larger models offered as cloud services.

TL;DR The short version

- A small language model runs on modest hardware such as one workstation, a phone or an edge device; there is no agreed size cutoff.
- Smaller models are cheaper and faster and can run where your data already lives.
- They are the wrong choice for open-ended reasoning or long, tangled documents, so test the smallest candidate on 50 real examples and move up only as needed.

Read online, listen to it, or change the text size: https://blogs.iictp.org/small-language-models-explained

Why a smaller model can be the better engineering choice

- **Cost and latency.** Fewer parameters means less memory and less compute per token, so answers arrive faster and cost less to serve.
- **Privacy and control.** A model that fits on your own hardware can run where the data already lives, including offline.
- **Predictability.** A narrow task with a fixed output shape, such as classification, extraction or routing, rarely needs a general-purpose giant.
- **Specialisation.** A small model tuned on your own task can match a larger general one on that task, though it will be worse outside it.

Where a smaller model is the wrong choice

Open-ended reasoning across many topics, long and tangled documents, and tasks you cannot describe ahead of time still favour larger models. A small model is a specialist, and the cost of using one for the wrong job is wrong answers delivered cheaply and quickly.

How to decide

1. Define the task and write 50 real examples with the answers you expect.
2. Test the smallest candidate model first, then move up only as far as the examples demand.

3. Check the failure cases, not the average. A model that is usually right and occasionally dangerous may be unusable.
4. Add a verification step for anything that matters. See applied AI: verify what the machine produces.

Running the chosen model yourself is covered in how to run an LLM on your own server.

> Model choice and local deployment sit in the Local LLM Practitioner credential.

Sources and further reading

1. IICTP. Local LLM Practitioner credential page. iictp.org/certifications/llmp.

Published by IICTP, an independent technical credential body, on 26 October 2026. The current version of this post is at https://blogs.iictp.org/small-language-models-explained. More guides: blogs.iictp.org.