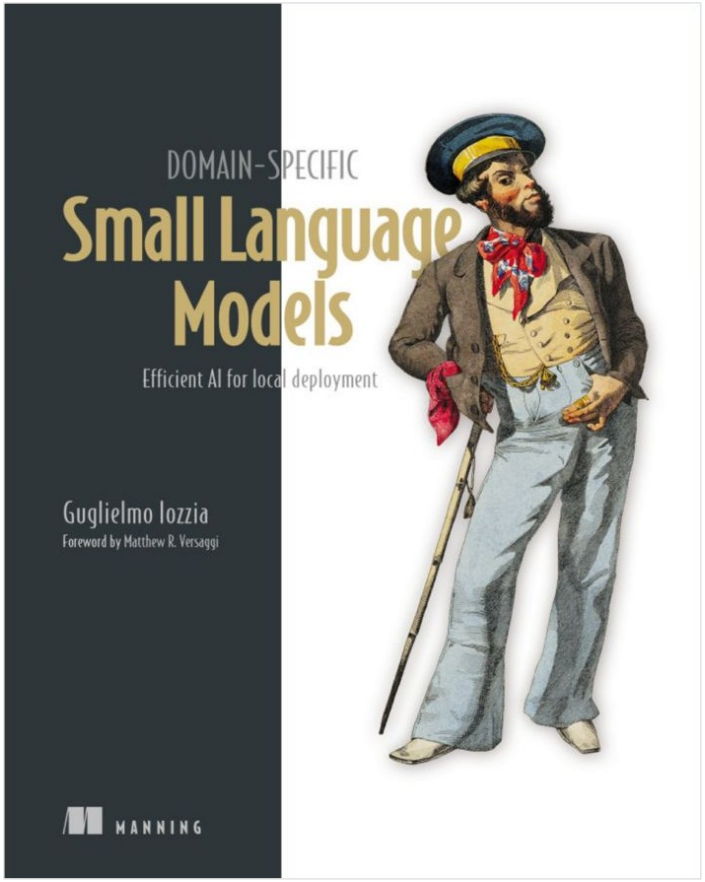
LIVE BOOK REVIEW · WEBINAR

Domain-Specific Small Language Models

Book Review and its Applicability to the Medical Domain



Dr. Wan Nor Arifin
Department of Biostatistics and Research Methodology
Universiti Sains Malaysia
wnarifin@usm.my / wnarifin.github.io

Thursday, 12 November 2026
8.00 pm MYT · Webex Online

Organized by Unit Kesihatan Digital USM, Malaysian Disease Modelling Experts Group and Malaysian Association of Epidemiology

CC BY SA

Outline

01 About the book

02 Why I read this book

03 Parts and chapters

04 Impressions



About

TITLE

Domain-Specific Small Language Models: Efficient AI for Local Deployment

AUTHOR

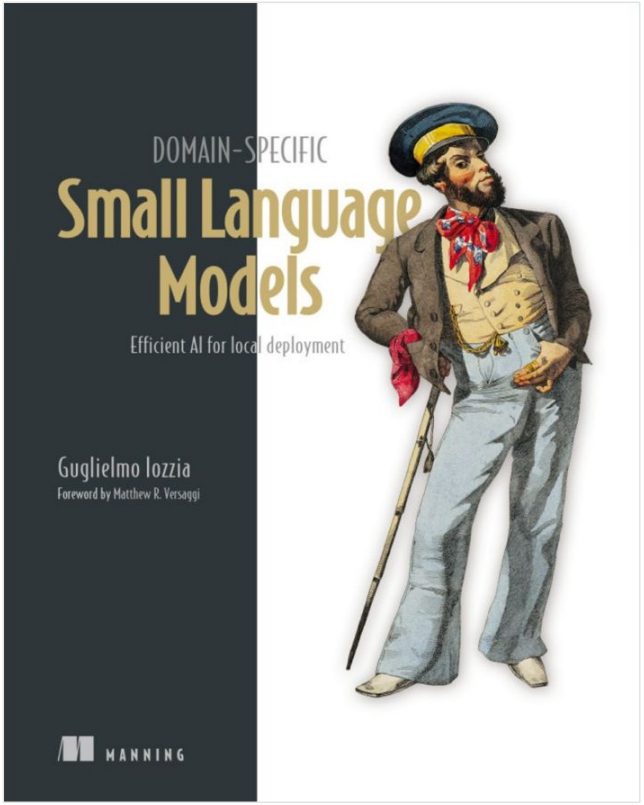
Guglielmo Iozzia

PUBLISHER

Manning

PUBLICATION DATE

May 2026



Cover image: Manning Publications

About the author

Guglielmo Iozzia

Director of ML/AI and Applied Mathematics
Merck & Co.

Open-source contributor: OptiLLM
An OpenAI API-compatible inference proxy to improve reasoning tasks



Image source: https://www.labcleanroomevent.com/speakers/guglielmo-iozzia

About the book

Summary

"... understanding techniques for improving inference performance and costs on pretrained and customized small language models (SLMs) through optimization and quantization, serving them through diverse API ecosystems, deploying them on diverse hardware (including your own laptop), and integrating them with other paradigms such as RAG and Agentic AI."

Target readers

- *"... **ML engineers and data scientists** interested in learning how to manage LLMs in the typical hardware-constrained environment ..."*
- *"... **tech leaders** motivated to understand how applying custom language models on corporate data could generate extra business value."*

About the book

The expected reader background is **relatively advanced**:

- General understanding of deep learning concepts
- Knowledge of the Transformer model's architecture and internals and the basics of their training process
- Intermediate Python language skills
- Hands-on experience with a deep learning framework (PyTorch preferable)
- Familiarity with the Google Colab environment
- Intermediate experience in software engineering
- Three or four years of experience as a machine learning engineer or data scientist



Why?

Diagnostic support

Text LLM, image VLM, multimodal MLLM → **differential diagnoses**

Workflow optimizer

Documentation, discharge summaries, structured data extraction, evidence aggregation → **simplify repetitive, complex tasks**

Patient

Health information chatbots, etc. (but a virtual doctor? I don't think that is ethical!)

The right view: an LM is an **assistant**, not a replacement for you!

Why?

Is it ethical to expose patients' data to LLM service providers?

They are POWERFUL, but ...

Locally deployed LLMs

- Better protect patients' data
- Customization to local context

Resource constraint

- Specialized **small language models (SLMs)**

MOTIVATION

Why?

General

- I have been working with LLMs since last year for research purposes
- I have been experimenting with SLMs that fit on my GPU, using Ollama
- Their potential in medicine in the coming years

Specific

- To learn the details of fine-tuning and adapting SLMs for specific use in public health

Book outline

Part 1 First steps

1 Small language models

Part 2 Core domain-specific LLMs

2 Tuning for a specific domain
3 End-to-end transformer fine-tuning
4 Running inference
5 Exploring ONNX
6 Quantizing for your production environment

Part 3 Real-world use cases

7 Generating Python code
8 Generating protein structures

Part 4 Advanced concepts

9 Advanced quantization techniques
10 Profiling insights
11 Deployment and serving
12 Running on your laptop
13 Creating end-to-end LLM applications
14 Advanced components for LLM applications
15 Test-time compute and small language models

Part 1: First steps

An SLM is a language model with **fewer than 10 billion parameters**

SLMs are ...

- Small, local, fast, energy efficient
- Customizable for private data and domain-specific tasks
- Open weight, transparent

Risks of generalist LLMs

- Patients' data are their "food": **confidentiality risk**
- "Personality" shifts between model versions: **change in expected responses**
- Lack of transparency: training details are **corporate secrets**

IMPRESSION

This is a **must-read** chapter.

Part 2: Core domain-specific LLMs

Ch 2 A code walk-through, from **data preparation to RAG and fine-tuning**. Full fine-tuning and PEFT (via LoRA) are demonstrated.

Ch 3 Implements the fine-tuning technique from Chapter 2 to build a fine-tuned **Manim code generator**. Introduces **hyperparameter tuning** and **evaluation** of the generated responses.

Ch 4 Technical overview of estimating the **hardware and monetary costs** of SLM inference, e.g., memory requirements of an LLM, KV caching.

Ch 5–6 Center on the **ONNX** framework and **quantization**. ONNX is an open format for cross-platform portability of ML models: a model's weights can be quantized, optimized and run on many platforms. It is an alternative to GGUF.

IMPRESSION

Readers may focus only on **Chapter 2**, and the **hyperparameter tuning and evaluation** parts of Chapter 3.

CHAPTERS 7–8

Part 3: Real-world use cases

CHAPTER 7

Generating Python code

CHAPTER 8

Generating protein structures

Examples of running LLM inference for Python code and protein structure generation, and how to **optimize the inference via ONNX and quantization**.

IMPRESSION

These chapters can be skipped and read later when needed.

Part 4: Advanced concepts (1/2)

Covers advanced concepts in quantization; LLM deployment, serving and inference; building around the model with RAG and agents; and test-time compute.

Ch 9 Advanced compression: **FlexGen** offloads and compresses large LLMs into RAM and disk; **SmoothQuant** is a specialized quantization that better handles outliers; **BitNet** is a 1-bit Transformer architecture with remarkably reduced size, cost and energy consumption.

Ch 10 Specific to the ONNX framework: **profiling and optimizing** ONNX models.

Ch 11 **vLLM** and **FastAPI** for serving LLMs via REST API; **MLC LLM** and **MLLM** for on-device deployment and inference (e.g., Android smartphones).

Ch 12 Options for running SLMs locally on a personal PC: **Ollama**, **LM Studio** and **Jan**.

Part 4: Advanced concepts (2/2)

Ch 13 Analogy: an SLM is the **engine**; a complete **car** needs other components. Building around the SLM using **RAG** and **agents**.



Ch 14 Expands on Chapter 13: **GraphRAG**, combining RAG with AI agents, and SLM **history/memory management**.

Ch 15 Introduces **test-time compute** and how it relates to reasoning models.

IMPRESSION

Chapters 11–15 are especially interesting and can be read for an overview of the concepts: the practical application and deployment of fine-tuned SLMs.

General impressions

The book covers **more than I expected**; I came for fine-tuning.

Most chapters are technical. We will need a **multidisciplinary technical team** to implement SLMs.

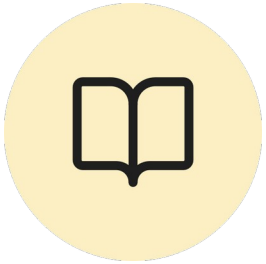
A good read to get an overview of what is required for the successful use of SLMs in the medical domain: **it's not just about getting a GPU!**

Issues

The main issue: **some code in the book did not work**, possibly due to updates in the packages used.

Manning's **liveBook** is helpful: its AI companion feature helps to correct some errors in the code.

The code in the book's **GitHub repository differs** from the published book in places: the repository provides extended code not available in the book, e.g., Section 2.4 on GitHub includes **evaluation of the generated content**, which is absent from the book.



Thank you! Happy reading!

Dr. Wan Nor Arifin
wnarifin@usm.my / wnarifin.github.io

Live Book Review Webinar · Thursday, 12 November 2026

