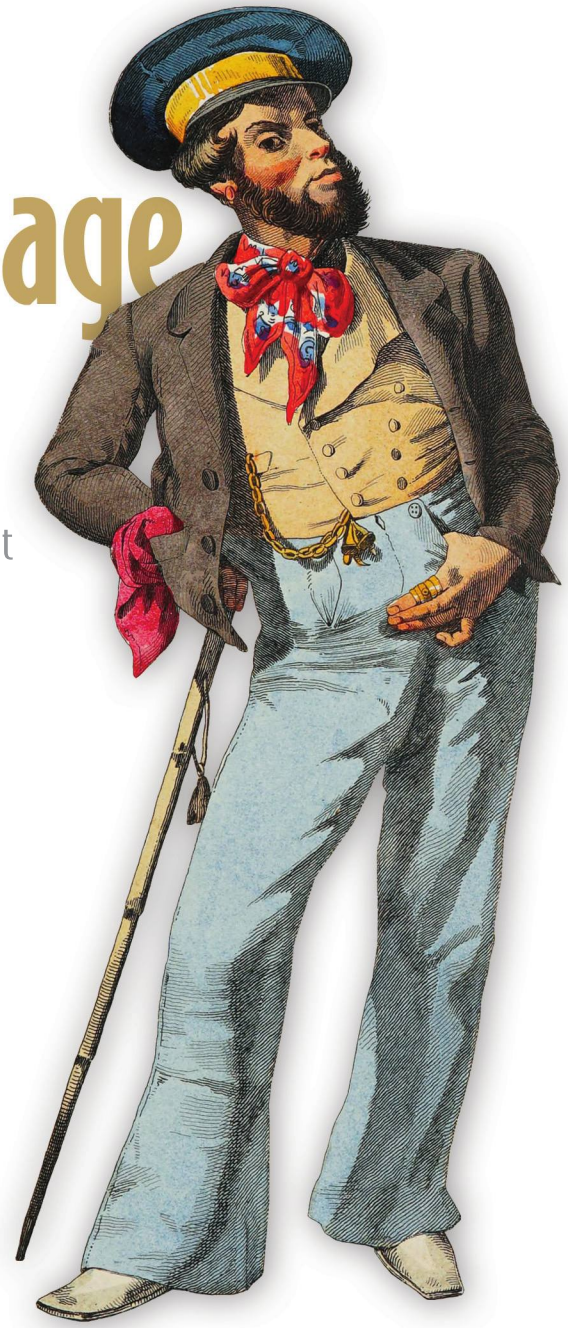
DOMAIN-SPECIFIC

Small Language Models

Efficient AI for local deployment



Guglielmo Iozzia

Foreword by Matthew R. Versaggi

MANNING

Domain-Specific Small Language Models

Domain-Specific Small Language Models

Efficient AI for local deployment

Guglielmo Iozzia

Foreword by Matthew R. Versaggi

MANNING
Shelter Island

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity.

For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

© 2026 Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

The author and publisher have made every effort to ensure that the information in this book was correct at press time. The author and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause, or from any usage of the information herein.

Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Development editor:	Dustin Archibald
Technical editor:	Riccardo Mattivi
Review editor:	Angelina Lazukić
Production editor:	Aleksandar Dragosavljević
Copy editor:	Andy Carroll
Proofreader:	Melody Dolab
Technical proofreader:	Bin Hu
Typesetter:	Tamara Švelić Sabljić
Cover designer:	Marija Tudor

ISBN 9781633436701
Printed in the United States of America

brief contents

contents

foreword

It is truly an honor to receive a request to write a foreword for a book like this, in this particular space, at this specific time in history, and by an author who I believe will make significant contributions to this industry during his tenure. That book is *Domain-Specific Small Language Models*, the space is generative AI, the time in history is the tremendous evolutionary acceleration of AI, and the author is Guglielmo Iozzia.

In an era when AI discourse is dominated by ever-larger frontier models and eye-catching demos, this book takes a refreshingly pragmatic stance: it is about making language models actually work for your domain, your data, and your hardware constraints. It is written with the practitioner's learning curve squarely in mind, moving from foundational concepts to hands-on experience so that by the time you turn the last page, you will not only understand why small, domain-specific language models matter, you will have run them, tuned them, optimized them, and deployed them on real-world infrastructure.

What distinguishes this work is its relentless focus on the realities of distributed and edge AI. Rather than treating GPUs in the cloud as an infinite resource, the author leans into the constraints most organizations actually face: tight budgets, on-prem clusters, laptops, and even mobile or embedded devices. From the opening chapters on the definition and role of small language models (SLMs), it becomes clear that the goal is not to worship scale, but to show how carefully chosen, well-tuned models in the 100M to 10B parameter range can deliver outsized business value when pushed as close as possible to where data is generated and decisions are made. The walkthroughs of ONNX, quantization techniques, and hardware-aware optimizations anchor the discussion in concrete practices that materially reduce latency and cost at the edge while preserving useful accuracy.

This is also an excellent grounding text for anyone trying to navigate the LLM-versus-SLM decision space with intellectual honesty. The early chapters provide a clear, hype-free tour of the Transformer architecture, training paradigms, open source ecosystems, and the tradeoffs between generalist closed LLMs and domain-specific models. From there, the book steadily builds into a toolkit: data preparation, classic fine-tuning, PEFT methods such as LoRA, and retrieval-augmented generation are all presented with enough depth and runnable code that you can reproduce, adapt, and extend the examples for your own environment. The result is a rare combination of conceptual clarity and practical guidance that serves both as an introduction for those new to the field and as a structured reference for experienced practitioners formalizing their approach.

Perhaps most compelling is the way the book repeatedly returns to domain reality. Coding assistants are explored not as abstractions but via end-to-end fine-tuning of GPT-2 on Manim code, complete with hyperparameter search and evaluation workflows that mirror production-grade ML engineering. Chemistry and life sciences are treated with equal seriousness, from protein and antibody sequence generation to discussion of validation workflows and specialized tooling, illustrating how SLMs unlock high-value use cases in regulated, data-sensitive industries. Later chapters connect these capabilities to modern application patterns—RAG pipelines, agentic systems, deployment and serving patterns, and running on commodity hardware—so that readers finish with a mental model that spans from theory to architecture to code.

If you are serious about applying generative AI beyond glossy prototypes—especially in domains where data is sensitive, regulation is strict, or infrastructure is constrained—this book is worthy of your time. It will not just tell you that small language models are important; it will show you, step by step, how to build, adapt, and ship them responsibly in the environments where they matter most.

—Matthew R. Versaggi,
White House Presidential Innovation Fellow in AI, Special cohort of '24,
former senior director of artificial intelligence and
cognitive technology at Optum Technology/United Healthcare

preface

In early 2022, after finally having the time to complete reading the "Attention is all you need" paper, I started getting curious about the Transformer architecture and the potential great use cases in diverse industries (and in life sciences in particular, because that is the field where I work) for this new kind of neural network. One concern I had at that time was the concrete risk that such technology could quickly become a prerogative of the large tech organizations that could afford the vast computational resources necessary to train and execute these models.

Then, inspired by the "run Doom on anything" challenge (a popular trend for software engineers to find ways to optimize the source code of that 1993 videogame to run on any sort of device), I started thinking about ways to optimize small Transformer models, trained on domain-specific tasks and data, so they could be deployed and executed in hardware-constrained environments. In June of the same year, I did an in-person hands-on workshop on this subject at the ODSC Europe conference in London, which got a lot of interest from ML engineers at the event.

When Manning Publications contacted me a few months later, asking me to write a book on this topic, I didn't think twice about accepting. A few weeks after I started writing the book, OpenAI released the ChatGPT service to the public, powered at that time by their GPT-3.5 model, and other large tech organizations, such as Google, Anthropic, and X.AI, soon joined the race to build larger, closed source generalist models to try to win the consumer market for generative AI.

But the open source community didn't just watch this happen: lots of initiatives involving open architectures and weights, and techniques for optimize the training and execution of language models, started being proposed. It was the right time to write a book on this topic and provide ML practitioners with a centralized source of information about what capabilities are available beyond the commercial offerings, for specific tasks, and in heavily regulated industries.

acknowledgments

I want to say thanks first of all to Jonathan Gennick, who reached out to ask if I was interested in sharing my experience on this topic through a book and for helping me through the nontrivial process of building the book proposal in such a way that it might be approved by the publisher. Thanks to Jonathan, and Manning Publications in general, for the support I received when, in the middle of writing, I had to stop for some months to address some urgent personal matters. Thanks to Dustin Archibald for all his precious feedback on making this book better. Thanks to Riccardo Mattivi, a senior AI and Machine Learning engineer at Integral Ad Science (IAS), who previously led AI engineering and Machine Learning initiatives at Mastercard, for serving as a technical editor on the book. And thanks, of course, to all the other Manning people who assisted during the different phases of this book, from inception to production.

To all the reviewers, Ajay Tanikonda, Alessandro Negro, Anand Nair, André Claudino, Andrew R. Freed, Andrey Belov, Carmine Giardino, Chandrashekar Konda, Clyde Kallahan, Dario Morelli, Dmitrii Volkov, Dmitry Kolodezev, Elif Ozkirimli, Fabio Montagna, Giampiero Granatella, Gregorio Piccoli, Harikrishnan Muthukrishnan, Hilde Van Gysel, Ho Chek Hui, Indraneel Rao, Karl-Gustav, Lola Vicente, Lukasz Kupka, Manchor Ko, Manish Jain, Manju Rupani, Marco Sassarini, Maxim Volgin, Mikael Dautrey, Muhammad Saqib, Nicolas Bievre, Prasad Rao Thota, Praveen Nair, Raj Kumar, Rani Sharim, Richard Vaughan, Sam Prakash Bheri, Samuel Lawrence, Saurabh Aggarwal, Sebastien Tardif, Shailesh Thakur, Shreesha Jagadeesh, Simone Sguazza, Srivathsan Srinivas, Surya Kasturi, Teo Argentieri, Thomas Tellner, Vibhas Zanpure, Walter Alexander Mata López, and Xin Hu, your suggestions helped make this a better book.

about this book

The focus of this book is on 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. All these concepts are explained in depth and come with complete source code examples. You'll learn to minimize the computational horsepower their models require while retaining high–quality performance times and output.

While a few examples presented in this book describe how to preprocess the data for training/test, and PEFT (parameter-efficient fine tuning) techniques are also introduced, this book doesn't focus on training and data-preparation techniques.

Who should read this book

This book is first for ML engineers and data scientists interested in learning how to manage LLMs in the typical hardware-constrained environment that their company budget allows for. But it is also for tech leaders motivated to understand how applying custom language models on corporate data could generate extra business value.

The minimally qualified reader should have the following skills and knowledge:

- 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

How this book is organized: A roadmap

This book has 4 parts with 15 chapters.

Part 1 introduces readers to small language models (SLMs):

- Chapter 1 explains the use cases for specialized SLMs and their pros and cons when compared to generalist large language models.

Part 2 covers the core topics of domain-specific SLMs:

- Chapter 2 shows some examples of data preparation for fine-tuning Transformer architectures and for RAG, and it introduces the concept of parameter-efficient fine-tuning (PEFT).
- Chapter 3 presents an end-to-end example of tuning an SLM on a specific task (the generation of Python code to render/animate mathematical formulas, starting from natural language). It also addresses some validation strategies for the model's output.
- Chapter 4 illustrates several types of content generation with SLMs, shows how to spot areas for improvement at inference time (in terms of cost savings and performance), and suggests some techniques for optimizing computational power consumption.
- Chapters 5 and 6 are about the ONNX format and ONNX Runtime, SLM quantization on CPU or GPU, and quantization strategies using the ONNX standard and alternative methods.

Part 3 focuses on real-world examples of the concepts introduced in parts 1 and 2:

- Chapter 7 presents some use cases of generating Python code with SLMs and optimizing SLMs to run on constrained hardware.
- Chapter 8 presents some SLMs pretrained to address tasks in chemistry, drug discovery, and materials science.

Part 4 discusses more advanced SLM concepts:

- Chapter 9 details advanced quantization techniques.
- Chapter 10 explains how to transform raw ONNX profiling data into insights to improve model optimization.
- Chapter 11 takes a deep dive into options for deploying and serving SLMs across diverse environments, including Android devices.
- Chapter 12 walks through multiple options for serving and running SLMs locally on a laptop and using a graphical interface.

- Chapter 13 is about using SLMs as part of more complex systems, such as RAG or agentic AI.
- Chapter 14 expands on the concepts discussed in chapter 13, introducing GraphRAG and agentic RAG.
- Chapter 15 is about test-time compute for SLMs, and it ends the book with an end-to-end example of turning an SLM into a reasoning model using GRPO.

About the code

This book contains many source code examples to give you hands-on experience with the concepts explained in the various chapters. To improve readability, the book often presents only the essential lines of code. Source code is formatted in a `fixed-width font like this` to distinguish it from ordinary text. Sometimes code is also presented **`in bold`** to highlight code that has changed from previous steps in the chapter, such as when a new feature adds to an existing line of code.

In many cases, the original source code has been reformatted; we've added line breaks and reworked indentation to accommodate the available page space in the book. In rare cases, even this was not enough, and listings include line-continuation markers (➥). Additionally, comments in the source code have often been removed from the listings when the code is described in the text. Code annotations accompany many of the listings, highlighting important concepts.

The complete examples, in the form of Colab notebooks, are available for download from the Manning website at https://www.manning.com/books/domain-specific-small-language-models and from GitHub at https://github.com/virtualramblas/Domain-Specific-Small-Language-Models.

liveBook discussion forum

Purchase of *Domain-Specific Small Language Models* includes free access to liveBook, Manning's online reading platform. Using liveBook's exclusive discussion features, you can attach comments to the book globally or to specific sections or paragraphs. It's a snap to make notes for yourself, ask and answer technical questions, and receive help from the author and other users. To access the forum, go to https://livebook.manning.com/book/domain-specific-small-language-models/discussion. You can also learn more about Manning's forums and the rules of conduct at https://livebook.manning.com/discussion.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the author can take place. It is not a commitment to any specific amount of participation on the part of the author, whose contribution to the forum remains voluntary (and unpaid). We suggest you try asking the author some challenging questions lest his interest stray! The forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

about the author

Guglielmo Iozzia is a Director of ML/AI and Applied Mathematics at Merck & Co. He studied Electronic and Biomedical Engineering at the University of Bologna, has an extensive background in Software and ML/AI Engineering applied to real-life use cases across different industries, such as Biotech Manufacturing, Healthcare, Cloud Operations and Cyber Security, and is a lifelong learner. He strongly believes that real progress in AI can happen only through Open Source.