

AI Development English

Participant workbook: jargon, dialogues, technical updates, eval readouts, and AI-development discussion practice

Audience: advanced ESL learners working in AI development or AI-adjacent technical roles

Focus: high-level professional English for AI development teams, including technical vocabulary, engineering discussion patterns, research/product tradeoffs, evaluation language, risk communication, and realistic workplace dialogue.

Designed for advanced ESL learners who already work with software, data, or AI systems and need field-specific fluency rather than basic grammar instruction.

Teaching stance: AI language changes quickly. Teach learners to ask precise clarification questions, define terms in context, and distinguish research claims, implementation details, benchmark results, and product promises.

How to Use This Workbook

This workbook helps you sound precise in AI-development conversations. The goal is not to use more jargon. The goal is to use the right term, define it when needed, ask useful questions, and explain technical risk in English that your team can act on.

Your starting point

- Which AI-development conversations are hardest for you: standups, design reviews, eval readouts, customer calls, incidents, or executive updates?
- Which terms do you understand when reading but avoid when speaking?
- When you disagree technically in English, do you become too vague, too quiet, too detailed, or too blunt?
- What is one AI failure you recently had to explain?

AI Stack Language

Layer	Useful verbs	Example sentence
Data	ingest, clean, label, split, anonymize	We need to check whether the eval set leaked into training.
Retrieval	chunk, embed, retrieve, rerank, ground	The retriever found similar chunks but missed the answer-bearing section.
Prompt	specify, constrain, format, include, truncate	The prompt does not specify what to do when evidence is missing.
Model	generate, refuse, summarize, classify, infer	The model follows the format but over-refuses safe requests.
Serving	cache, batch, stream, throttle, fallback	P95 latency increased after we added the reranker.
Evaluation	measure, slice, regress, calibrate, inspect	The average improved, but the high-risk slice regressed.

Practice Pages

Module 1. Speaking the AI Development Stack

AI teams use a layered vocabulary: model, data, prompt, retrieval, tools, serving, evaluation, monitoring, and product experience. Learners need to locate a problem in the stack before they can discuss it clearly.

What you should be able to do

- Name the layer where an AI issue occurs.
- Distinguish model capability from application behavior.
- Use concise stack language in standups and tickets.

Practice task

Situation

A bug report says, 'The AI is bad.' Rewrite it into a stack-specific issue with observed behavior, expected behavior, evidence, and likely owner.

Your response

Clarification questions

Final professional sentence

Module 2. LLMs, Transformers, Tokens, and Context

High-level AI communication often depends on explaining what the model sees: tokens, messages, context window, instructions, examples, retrieved text, and tool results.

What you should be able to do

- Explain tokens, context window, prompts, and attention in workplace English.
- Discuss prompt changes without implying that prompting is magic.
- Describe context limits, truncation, and cost/latency tradeoffs.

Practice task

Situation

Explain to a product manager why adding more retrieved context can increase cost and latency without improving answer quality.

Your response

Clarification questions

Final professional sentence

Module 3. Data, Datasets, Labels, and Leakage

AI systems are shaped by data quality. Teams need precise language for dataset splits, annotation guidelines, leakage, imbalance, representativeness, and privacy constraints.

What you should be able to do

- Describe a dataset and its limitations without overclaiming.
- Explain labels, ground truth, noisy labels, and inter-annotator agreement.
- Raise leakage and privacy concerns clearly.

Practice task

Situation

Write five clarification questions for a dataset that may have label noise, class imbalance, and privacy restrictions.

Your response

Clarification questions

Final professional sentence

Module 5. Fine-Tuning, Alignment, and Adaptation

Teams often confuse prompt changes, RAG, fine-tuning, adapters, supervised fine-tuning, preference tuning, and RLHF. The language goal is to recommend the right adaptation method for the problem.

What you should be able to do

- Differentiate prompt engineering, RAG, fine-tuning, LoRA/adapters, SFT, DPO, and RLHF at a practical level.
- Discuss when fine-tuning is not the right solution.
- Explain cost, data, evaluation, and maintenance tradeoffs.

Practice task

Situation

Choose between prompt update, RAG, fine-tuning, and product logic for a model that refuses too often in safe cases.

Your response

Clarification questions

Final professional sentence

Module 8. Safety, Security, Privacy, and Governance

AI teams must discuss risk precisely: hallucination, prompt injection, jailbreaks, PII, data retention, bias, harmful output, policy enforcement, audit logs, and human-in-the-loop review.

What you should be able to do

- Name safety and security risks without dramatic overstatement.
- Explain guardrails, validation, red-teaming, and monitoring.
- Escalate privacy or policy concerns with evidence.

Practice task

Situation

Write a safety escalation for a prompt-injection attempt found in a third-party webpage.

Your response

Clarification questions

Final professional sentence

Phrase Bank

Clarifying architecture

- Which layer do we think is failing: retrieval, prompt, generation, validation, or UI?
- Is this a model behavior issue or an application orchestration issue?
- Can we separate the model output from the wrapper logic?

Discussing data

- What is the source and coverage of this dataset?
- Do we have evidence of label noise or annotator disagreement?
- Could there be leakage between the training set and the eval set?

Reporting evals

- The overall metric improved, but one high-risk slice regressed.
- The sample size is small, so I would treat this as a warning, not a conclusion.
- We should inspect failure examples before making a release decision.

Explaining RAG

- The model did not have the right context, so the generation step was answering from weak evidence.
- The retriever found similar text, but not the text that actually answered the question.
- We need better chunking, metadata, or reranking before changing the model.

Pushing back

- I do not think fine-tuning is the first fix here; the failure looks like retrieval or policy logic.
- A benchmark win is not enough unless it transfers to our product eval.
- We should not ship until the severe failure slice is understood.

Customer-safe language

- The answer was unsupported by the available source material.
- We are investigating whether the issue came from retrieval, grounding, or validation.
- We have paused that workflow while we review the affected traces.

Personal Action Plan

Situation	Phrase or term I will practice	Evidence I used it well