EFSP Auxiliary ESL Curriculum

AI Development Jargon Field Guide

Quick reference for high-level ESL learners who need precise AI-development vocabulary and meeting language

Audience: advanced ESL learners in AI engineering, research, product, deployment, safety, and support

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 Jargon Well

- Use the term only when it locates the problem more precisely.
- Pair jargon with evidence: metric, trace, log, example, user impact, or source document.
- Define the term when speaking to product, support, legal, sales, or executives.
- Avoid vague AI blame. Name the layer and the next diagnostic step.

Nomenclature and Jargon

Teach these terms as working vocabulary. Learners should be able to define the term, use it in a sentence, ask a clarification question about it, and explain its business consequence.

Core model terms

Term	Working meaning
LLM	Large language model; a model trained to process and generate language-like sequences.
Transformer	A neural architecture based on attention mechanisms, common in modern language models.
Parameter	A learned numerical value in a model; not the same as an API parameter.
Checkpoint	A saved version of model weights at a point in training or fine-tuning.
Foundation model	A broadly trained model adapted to many downstream tasks.
Multimodal	Able to handle more than one data type, such as text, image, audio, or video.

Prompt and context terms

Term	Working meaning
Prompt	The instructions, examples, user request, and context given to a model.
System prompt	High-priority instructions that guide model behavior inside an application.
Few-shot	Including examples in the prompt to show the desired pattern.
Context window	The amount of input and generated text the model can consider in one request.
Token	A unit of text processed by the model; token count affects cost, context, and latency.
Temperature	A generation setting that affects output variability.

Retrieval and RAG terms

Term	Working meaning
Embedding	A vector representation used for similarity search, clustering, classification, and related tasks.
Vector store	A database or index for storing and searching embeddings.
Chunking	Splitting documents into retrievable pieces.
Reranker	A model or step that reorders retrieved results for relevance.
RAG	Retrieval-augmented generation: retrieve relevant context, then generate an answer using it.
Grounding	Tying model output to retrieved, cited, or verified source information.

Training and adaptation terms

Term	Working meaning
Fine-tuning	Updating model weights on task- or domain-specific data.
SFT	Supervised fine-tuning with input-output examples.
RLHF	Reinforcement learning from human feedback; training with human preference signals.
DPO	Direct preference optimization; preference tuning without a separate reward model in common workflows.
LoRA	Low-rank adaptation; a parameter-efficient fine-tuning method.
Adapter	A small trainable module inserted into or attached to a pretrained model.

Evaluation terms

Term	Working meaning
Eval	A test or evaluation suite for model or system behavior.
Benchmark	A standardized test used to compare systems, often imperfect for a product use case.
Golden set	Curated examples used repeatedly to test important behavior.
Regression	A behavior that gets worse after a change.
Pass rate	The percentage of eval cases meeting the success criterion.
LLM-as-judge	Using a model to evaluate outputs, usually with calibration and human review.

Production terms

Term	Working meaning
Inference	Running a trained model to produce an output.
Latency	How long a request takes to return a result.
Throughput	How many requests a system can handle in a period of time.
Batching	Processing multiple requests together for efficiency.
Streaming	Sending partial output to the user as it is generated.
Fallback	A backup behavior when the preferred path fails.

Safety and security terms

Term	Working meaning
Hallucination	A generated claim that is unsupported, false, or not grounded in the provided context.
Prompt injection	Untrusted input tries to manipulate model instructions or tool use.
Jailbreak	A prompt or interaction that tries to bypass safety constraints.
Guardrail	A control that detects, blocks, changes, or routes risky behavior.
PII	Personally identifiable information.
Red team	A structured effort to find failures, vulnerabilities, or unsafe behavior.

Common Meeting Moves

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.

Fast Contrast Pairs

Do not confuse	Working contrast
Prompting vs fine-tuning	Prompting changes instructions/context; fine-tuning changes model weights.
RAG vs training	RAG retrieves external context at inference time; training changes what the model has learned.
Benchmark vs product eval	A benchmark is general comparison evidence; a product eval tests your actual use case.
Hallucination vs retrieval miss	Hallucination is unsupported output; retrieval miss is failure to fetch needed context.
Latency vs throughput	Latency is one request's wait time; throughput is system volume over time.
Safety vs security	Safety reduces harmful outputs; security protects systems, data, tools, and permissions.

Source Orientation

- OpenAI API documentation: embeddings, tools/function calling, structured outputs, evals, and agent eval guidance.
- OpenAI Cookbook examples on evaluation and RAG workflows.
- Hugging Face documentation: Transformers, Tokenizers, PEFT, Evaluate, and Hub concepts.
- Google Machine Learning Glossary and Responsible AI glossary.
- Lewis et al., Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, NeurIPS 2020.