



Entwicklung eines Fine-Tuned-Modells zur automatisierten Erstellung von H5P-Inhalten im Bereich Cybersicherheit

Ahmad Alzohaile - Dawing Ronaldo Diefouo

Projektarbeit • Studiengang Informatik-Master • Fachbereich Informatik und Medien • 15.01.2026

Aufgabenstellung

Die Zielsetzung dieser Arbeit besteht in der Entwicklung eines sogenannten "Fine-Tuned-Modells", dessen Fokus auf der Erstellung automatisierter H5P-Inhalte im Bereich der Cybersicherheit liegt. Das Projekt ist als Teilprojekt des übergeordneten SCALE-C-Projekts zu begreifen.

Was ist H5P?

H5P ist ein ZIP-Paket, das interaktive Lerninhalte wie Quizze, Präsentation und vieles mehr enthält. Darin liegt die *content.json* Datei, die in unserem Projekt als Zielstruktur dient.

Finetuning

Der Begriff "Finetuning" bezeichnet der Prozess, ein Modell auf spezifische Aufgaben oder Anwendungsfälle anzupassen oder zu optimieren.

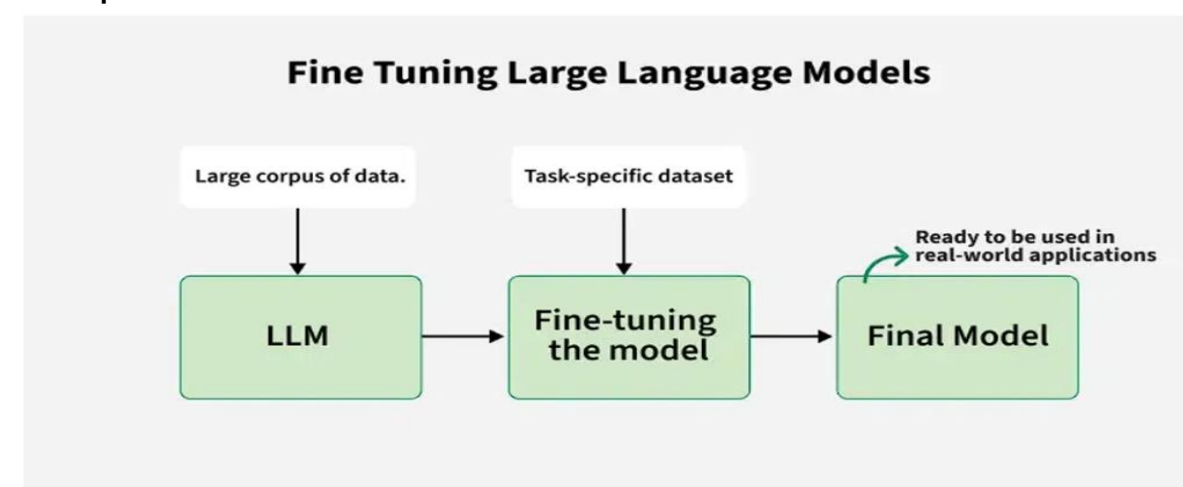


Abb. 1: LLM-Fine-Tuning [1]

Um der Fine-Tuning-Prozess zu realisieren, lässt sich hauptsächlich in fünf Schritte durchzuführen:

- 1) Modellauswahl: Mistral, DeepSeek, Gpt, Ollama, ...
- 2) Datenvorbereitung: Datensätze werden ins Trainingsformat gebracht
- 3) Auswahl der Fine-Tuning-Methode: Lora, Full-Finetuning, Instruction-Fine-Tuning
- 4) Training (Fine-Tuning)
- 5) Tests und Evaluation

Was unsere Arbeit angeht, haben wir erstens mit einem kleinen Modell, nämlich **tinylama** angefangen. **Tinylama** ist ein ein Sprachmodell mit **1.1B Parameter** und lässt sich lokal auf Computer trainieren, die nur über einen CPU verfügen. Als Ansatz zum finetunen haben wir uns für **Instruction-Fine-Tuning und LoRA** entschieden. Instruction-Fine-Tuning bestimmt die Art, wie das Modell auf User-Prompt reagiert. Und LoRA kümmert sich um das Training. Das heißt, welche Parameter angepasst werden sollen, und in welche Richtung. Und so bekommen wir am Ende ein Sprachmodell dessen Vorwissen nur teilweise überschrieben wurden.

Tools

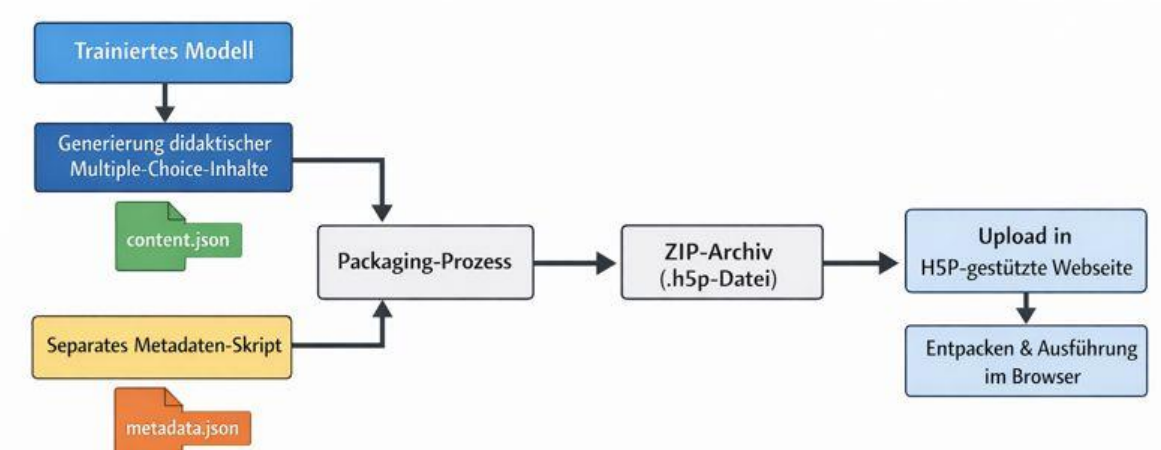
Um ein Sprachmodell zu trainieren, brauchen wir eine komplette modulare ML-Toolchain. Unter anderen wichtigen Tools haben wir:

- ❖ Hugging Face Transformer: das ist das Herzstück unseres Trainings, da es begleitet uns beim initialisieren des Modells, tokenisieren, trainieren, ...
- ❖ Hugging Face Datasets: Saubere Trennung zwischen rohen JSONL-Daten und tokenisierten Trainingsbeispielen.
- ❖ PEFT (Parameter-Efficient Fine-Tuning): Nur ein **kleiner Prozentsatz** der Modellparameter wird trainiert
- ❖ PyTorch
- ❖ Python mit IntelliJ IDE

Ergebnisse

Am Ende des Trainings ist das Modell in Lage didaktische Multiple-Choices-Inhalte zu erzeugen, die dann in einer Json-Datei gespeichert werden. Aber eine H5P-Datei besteht zusätzlich aus anderen Meta-Daten, die durch ein anderes Skript generiert und anschließend in einer separaten JSON-Datei gespeichert werden. Die beiden werden dann gezippt in einer gemeinsamen H5p-Datei. Diese Datei laden wir in eine H5P-gestützte Webseite hoch, die infolgedessen ausgeführt werden kann.

Abb. 2: Generierung und Packaging einer H5P-Datei



Generierung didaktischer Multiple-Choice-Inhalte durch ein trainiertes Modell, separate Erzeugung der für H5P erforderlichen Metadaten sowie anschließendes Packaging beider JSON-Dateien zu einer lauffähigen H5P-Datei, die in einer H5P-gestützten Webplattform ausgeführt werden kann.

Abb. 2: Generierung und Packaging einer H5P-Datei

Fazit

In dieser Arbeit wurde gezeigt, dass die Erstellung automatisierte H5P-Inhalte im Bereich der Cybersicherheit möglich ist. Genauer lassen sich H5P-Inhalte für die Cybersicherheit mithilfe moderner Fine-Tuning-Ansätze automatisiert generieren.

Quellen

- [1]: <https://medium.com/@khalidsabban/fine-tuning-large-language-models-with-lora-a-step-by-step-tutorial-on-kaggle-d8e4ff961057>
- <https://datasolut.com/feinabstimmung-llms-best-practice>
- <https://www.geeksforgeeks.org/deep-learning/fine-tuning-large-language-model-llm>
- [https://de.wikipedia.org/wiki/Fine-Tuning_\(K%C3%BCnstliche_Intelligenz\)](https://de.wikipedia.org/wiki/Fine-Tuning_(K%C3%BCnstliche_Intelligenz))
- OpenAI. (2026). ChatGPT