# Print Edition

Sunday 16 August 2026

EDITION NO. 24 · 1 PIECE · 4 PAGES

IN THIS EDITION



# Building an AI Text Detector From Scratch

BY SEBASTIAN RASCHKA, PHD · AHEAD OF AI · 15 AUGUST 2026

Substack recently launched its AI detector feature in the UI, which is super interesting.

Separately, lots of people asked me about interesting local do-it-yourself LLM projects as demos to show what small language models (SLMs) are capable of.

Putting one and one together, I thought it would be interesting to show how an AI detector can be implemented. I will also use it as a verifier to train a small language model to produce text that avoids detection. This is a small educational project for studying the limitations of AI detectors and exploring a verifier-based LLM application beyond regular reasoning models trained on math and code.

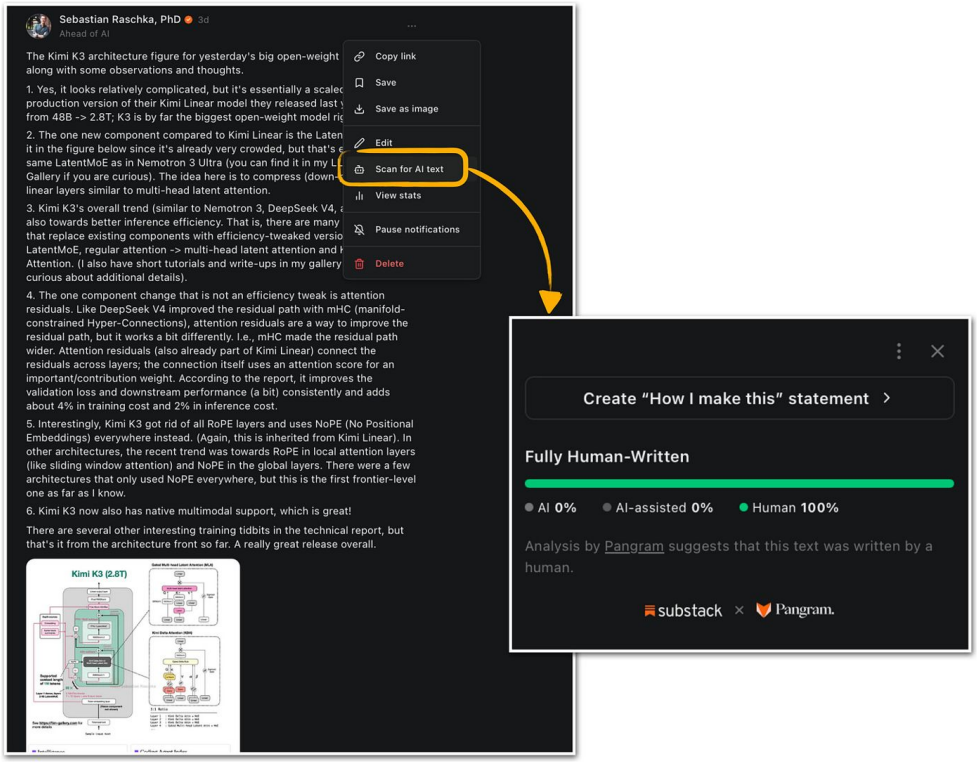


*Figure 1: Substack now features a built-in AI detector.*

So, as mentioned above, the intended goal of this tutorial is to explain how AI detectors work by building (a simple) one.

In practice, such a detector can be used to filter out spammy content, but also to potentially improve your personal writing without turning it into AI-generated text. For example, if you wrote a lengthy article and want to improve spelling and grammar, it is tempting (and actually useful) to use a grammar checker to polish it and improve readability. There are different services for that, including general-purpose LLMs like ChatGPT. However, this also runs the risk that these tools turn your writing, even though it's still your own writing, into something that is then overpolished and now sounds like AI and gets flagged as spammy content.

For example, with an AI checker, one could say, "Fix my grammar while ensuring that my text still scores 0% AI-generated."

Anyway, while we are building a fully functional checker here, the goal is to explain 1) how AI checkers (can) work and 2) use this as a case study for a more general topic on how to build a scorer or verifier that can be used with LLMs.

Disclaimer: AI checkers are essentially a cat-and-mouse game. AI checkers may learn to detect a certain pattern that is indicative of AI-generated content. Then, the next LLM may incidentally or deliberately not exhibit that pattern and avoid detection. The AI checker then has to be updated to detect said LLM, and so forth. Plus, it's also likely to encounter false positives (human written text flagged as AI-generated), but more on that later.

## Project goals

There are several goals of this project. The overarching goal is, of course, to illustrate how AI detectors work and show an applied end-to-end LLM project including evaluation, training, and local deployment for real-world use.

The outcome of this is an AI-detector API that can be used by humans and agents, and a user-friendly UI.


*Figure 2: Preview of the local browser interface developed later in this project. It returns a whole-text AI score and can also highlight the scores for individual text chunks.*

## Method overview

Here, we are going to develop a method similar to Pangram models, which, as far as I know, are behind Substack AI detection feature.

I wrote a short article about AI-text detection a while back in 2023: What Are the Different Approaches for Detecting Content Generated by LLMs Such As ChatGPT? And How Do They Work and Differ?

In essence, there are different ways to detect AI-written text, from supervised classifiers and perturbation-based probability tests to perplexity measures and watermarking.

In this tutorial, we will build a model that returns a 0-100 score. It's essentially a classifier with an estimated probability score. The probability score will denote how likely a text is AI-generated according to the classifier. (Or, to be precise the score is the classifier's estimated probability for the AI-generated class based on its training distribution. However, we shouldn't interpreted it as a general probability that the text was written by AI.)

For this, we are going to fine-tune a DistilBERT classifier (similar to what I described in one of my early Substack articles, Finetuning Large Language Models), but more details on that later when we get to that stage.

CONTINUES ONLINE · https://magazine.sebastianraschka.com/p/ai-detector-from-scratch