Reanimation Is All You Need

ZombieLLM - The Return of GPT-2 with a Mind of GPT-OSS-20B

Michal Wojewodzki Dominika Szyfter

Abstract

ZombieLLM is a mischievous experiment in model necromancy: we take GPT-2 XL (1.5B, 2019), give it a carefully supervised conversation with a modern open-weight giant (GPT-OSS-20B, 2025), and stitch the pieces together with DoRA finetuning, an alignment pass that matches representations, and a final personality tune so it talks like a helpful survivor rather than a lab report. The result is an offline-first, instruction-following model that runs on ordinary laptops and even cyberdeck-style Raspberry Pis, answers concisely by design, and carries just enough borrowed knowledge to be useful when the internet isn't an option. This paper explains why we did it, how we reanimated the corpse, what it can and cannot do, and where the sharp edges are. We are candid about limits: the model's base is still GPT-2, its context window is only 1,024 tokens, it was trained primarily on English instructions distilled from Dolly-15k and Alpaca prompts, and it can hallucinate. But it is responsive, consistent, and personable; it follows instructions; and in the grand tradition of good field tools, it "does one thing well": give short, direct answers without meandering reasoning traces. Our motto, like our outputs, is simple: **brains, but concise**.

Why reanimate GPT-2?

Large models are brilliant, heavy, and needy. They prefer modern GPUs and uninterrupted network access, neither of which is guaranteed in the real world (or during a hackathon demo or zombie apocalypse). We asked a stubborn question: could a classic, tiny model be coaxed into something practically useful if a modern giant mentored it long enough? GPT-2 XL is historically important and widely supported in runtimes such as transformers, llama.cpp, and Ollama backends that target CPUs. It also arrives with a harsh reality: by today's standards, it is under-equipped, with a short context window and a pretraining diet that ends in 2019. Rather than pretend otherwise, we embraced the limitation and reframed the goal. ZombieLLM is not trying to beat state-of-the-art. It's a semi-serious survival tool: portable, private, and just good enough to guide you through simple tasks when bigger models aren't available.

Anatomy of a reanimation

The pipeline begins with teacher-led distillation. We keep the questions from two classic instruction datasets (Dolly-15k and Alpaca) and ask the teacher-GPT-OSS-20B-to produce strict, final-only answers. The Harmony chat template keeps the teacher on a short leash: no bullet lists, no chain-of-thought, no role tags, and no hedging. A set of sanitizers removes classic verbosity tics ("As an AI…", "Certainly!") and trims any analysis channels that sneak through. The output is a pair of distilled instruction–response corpora shaped for compact models that don't have the luxury of long, wandering explanations.

We then fine-tune GPT-2 XL with TRL using DoRA in bf16. Packing, masking, and a completion-only collator ensure the student learns to generate just the response region in our prompt template. Representation-level knowledge distillation follows: we introduce simple projection heads that map the teacher and student final-layer activations into a shared space and nudge the student toward the teacher's "shape of thought" using a cosine loss. It's a small, pragmatic alignment step that behaves like an orthotic: not dramatic, but it helps the student stand straighter.

Finally, because tools need bedside manner, we give the model a short personality tune. A small mix blends practical survival Q&A with a consistent voice-calm, direct, a little wry. A brief booster pass stabilizes tone without eroding factual behavior. The stitched-together model is exported in Hugging Face format and then converted to GGUF for llama.cpp and Ollama, because a zombie should travel light.

The Zombie Equation

The reanimation formula looks like this:

ZombieLLM = GPT-2 XL + SFT(dollypaca ← GPT-OSS-20B final-only) + KD(rep-align) + DoRA(persona+survival) − hedging − chain-of-thought.

Read it like a field recipe: start with an old body, feed it clean, concise answers from a big brain, align the posture, add a personality coat, and subtract the fluff. The arithmetic is tongue-in-cheek, but the effect is real: responses become shorter, more decisive, and more consistent, which is exactly what a small model can sustain.

Data, guardrails, and the art of saying less

Our distilled datasets keep the original instructions and replace the answers with GPT-OSS-20B's concise finals. The Alpaca-cleaned sample we used is roughly thirteen thousand items, and the Dolly15k pass stays in that same ballpark. Every item uses the same text prompt template at finetune time and at inference, so the model's training diet matches the way you call it. Because we want the model to be unflappable on small devices, we train it to avoid verbosity by construction. There is no chain-of-thought in the dataset; you will not get one at inference. When it is unsure, we encourage it to decline. When a context is provided, it should stay within the four corners of that context.

We also curate a small persona and survival set (questions from moremilk/CoT_Reasoning_Bushcraft_Survival with GPT-OSS-20b distilled answers). The survival data nudges the model toward practical, non-dramatic answers in domains where tiny models can genuinely help: checklists or short procedures. The persona set makes the voice a little more human without tipping into roleplay; think of it as a tone profile rather than a character.

Training notes you can actually reuse

Everything is scripted in notebooks that prefer PyTorch-only mode and disable accidental TensorFlow imports. We run bf16 and we use gradient checkpointing to squeeze onto modest GPUs during training. The base SFT uses a completion-only collator that masks everything before the "### Response:" marker. The KD stage freezes the teacher, computes student hidden states, and pools both sides over the response span; projection heads reduce the mismatch in hidden sizes and keep the math cheap. A cosine similarity penalty with a modest weight is added to the usual cross-entropy loss. It is a single-epoch nudge rather than an ideology.

On the tooling side, we normalize tokenizer pad tokens, fix padding to the right for GPT-2 during SFT, and handle left-padding in the teacher path when doing batched Harmony renders. The final artifact includes a Jinja chat template directly in the tokenizer config so that multi-turn prompts still land in the same structure the model trained on.

Limits

ZombieLLM is still GPT-2 at heart. Its pretraining knowledge is limited and dated, and while the distilled answers from GPT-OSS-20B add breadth, they mostly reflect the style and scope of Dolly and Alpaca prompts. The model will hallucinate if pushed outside familiar territory or teased into open-ended speculation. It lacks the modern scaffolding that props up larger models: no tool use, no retrieval by default, no long-term memory. Its context window is short; copying pages of text into the prompt will not go well. It is English only. Biases present in the sources may surface in outputs. You should verify anything, and you should never use this model for medical, legal, financial, safety-critical, or high-stakes decisions. We mean that. In the unavoidable words of the responsible AI section: **research use only**.

Licensing follows the data. Because core training relied on CC BY-NC 4.0 sources (e.g., Alpaca-derived material) and distilled answers are integrated with those prompts, the released weights are **CC BY-NC 4.0**. That means no commercial use. If you need a commercial-friendly variant, you will have to rebuild with compatible data. Transparency beats ambiguity here.

How to run the zombie without waking the neighbors

The model is available as a standard Hugging Face directory, so you can load it with transformers and a simple text-generation pipeline. The chat template embedded in tokenizer_config.json mirrors the training prompt, so using chat-style messages will produce the same completion-only behavior: an “Instruction,” an optional “Context,” and a “Response” that doesn’t meander. For those who prefer packaged experiences, there is an Ollama build that puts a bow on top; it runs locally and does not phone home. However you deploy it, keep generation settings conservative: low temperature, modest top_p, and firm controls. The zombie performs best when you don’t ask it to improvise jazz.

Evaluation, qualitatively honest

We did not chase leaderboard scores that privilege scale. Instead, we looked for three practical behaviors: does the model follow instructions without dithering, does it stay inside a provided

context without inventing outside facts, and does it keep answers compact enough to be useful in tight interfaces? On these axes, ZombieLLM is satisfying. It also adopts the survivalist tone consistently after the booster pass.

Acknowledgments and related work

We stand on generous shoulders. GPT-2 XL gives us a venerable backbone. Dolly-15k and Alpaca provided instruction framing that the community understands and can reproduce. GPT-OSS-20B offered an open-weight teacher capable of producing consistent, final-only answers under a Harmony template. TRL, PEFT, and the DoRA trick made our finetunes lightweight and cheap. If this project feels like a well-stocked workbench rather than a moonshot, good-reanimation is a craft.

Closing, with one last candor check

This model has a personality and a sense of humor, but it is not alive; it understands English well, but it does not understand *reality*. Treat it as a compact assistant that answers briefly and behaves itself in small contexts. Verify anything that matters. And if you forget everything else, remember the field rule that guided every design choice here: **brains, but concise.**

References

1. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8):9 (2019).
2. OpenAI. *gpt-oss-120b & gpt-oss-20b Model Card*. arXiv:2508.10925 (2025). https://arxiv.org/abs/2508.10925
3. Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., Xin, R. Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM. *Databricks Blog* (2023). https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm
4. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T. *Stanford Alpaca: An Instruction-following LLaMA model*. GitHub repository (2023). https://github.com/tatsu-lab/stanford_alpaca
5. Wesney, M. R. *CoT_Reasoning_Bushcraft_Survival_Dataset*. Hugging Face (2025). https://huggingface.co/datasets/moremilk/CoT_Reasoning_Bushcraft_Survival
6. hardrave/zombiellm - Contains the ZombieLLM weights, tokenizer, and configuration files. https://huggingface.co/hardrave/zombiellm
7. ZombieLLM on Ollama - ready-to-run build. https://ollama.com/hardrave/zombiellm

8. hardrave/zombiellm (GitHub) - Jupyter notebooks for data distillation, supervised fine-tuning, and knowledge distillation. https://github.com/hardrave/zombiellm/
9. ZombieLLM datasets collection - Distilled datasets (Dolly, Alpaca-cleaned, Bushcraft_Survival, Zombie Persona) used for SFT and KD. https://huggingface.co/collections/hardrave/zombiellm-datasets-68c24e950f3b8559f11bf352