

llama.ttf

llama.ttf is a font file which is also a large language model and an inference engine for that model.

Ehm, what?

llama.ttf is a font file which is also a large language model and an inference engine for that model.

Why?

How?

The font shaping engine [HarfBuzz](#), used in applications such as Firefox and Chrome, comes with a [Wasm shaper](#) allowing arbitrary code to be used to "shape" text.

In particular, this "arbitrary" code could in principle be an entire LLM inference engine with trained parameters bundled inside, relying on treating text containing magic symbols for fake "ligatures" to initialize the LLM and use it to generate text.

It could also in principle be an entire LLM inference engine ([Llama](#) in our case, hence the name) except instead of only being in principle it's what this is.

At the end of the day, what this means is that you can just use the font to run the LLM and e.g. get text generation in *any* Wasm-enabled HarfBuzz-based application; your favorite text editor/email client/whatever without having to wait for the vendor to include the "Copilot"-like features that everyone is rushing to implement these days. And everything runs completely locally. So perhaps this silly hack is in fact a billion dollar idea!? This also means that you can use your font to chat with your font.

Okay show me

So in case that doesn't make sense (after all, does it?), here's an attempt to make it make more sense:

Using llama.ttf to eradicate writer's blocks once and for all



Skip to [6:09](#) if you just want to see the font in action.

Usage

Just download [llama.ttf](#) (60 MB download, since it's based on the 15M parameter TinyStories-based model demoed above) and use it like you would any other font. Use it somewhere where HarfBuzz is used and built with Wasm support. The simplest way to experiment with this is probably to build [HarfBuzz](#) with `-Dwasm=enabled` and build [wasm-micro-runtime](#) with `-DWAMR_BUILD_REF_TYPES=1`, then add the resulting shared libraries, `libharfbuzz.so.0.60811.0` and `libiwasm.so` to the `LD_PRELOAD` environment variable before running a HarfBuzz-based application such as `gedit` or `GIMP`; no recompilation of the applications is required.

More fun with fonts

If this didn't seem pointless enough, here are some other weird things people have done with fonts and LLMs:

- tom7's Super Metroid speedrun documentation-inspired language for formatting text nicely with LLMs: <https://www.youtube.com/watch?v=Y65FRxE7uMc>
- Erk's presentation on programmable fonts with HarfBuzz-Wasm, very much an inspiration for this one: <https://www.youtube.com/watch?v=Ms1Drb9Vw9M>
- A completely playable Pokémon-inspired game inside a font: <https://www.coderelay.io/fontemon.html>