# Capturing the Soul of Toby Plaus

By Jack Brassil

# Table of Contents

**Table of Contents** I

**Intro and Goal** 1

**What is Romance in Italy?** 3

**Technical Stack** 5

**Some Questions for Toby** 6

**Fine-Tuning: The Finn Problem** 7

**Overfitting: The Wingdings Problem** 9

**Merging: The Repetition Problem** 11

**RAG: The Hannah Problem** 12

**Future Work** 13

**Conclusion** 14

**Citations** 15

# Intro and Goal

Previously, I did a project where I analyzed the performance of 3 different classification models against a dataset of news articles. As I completed that project, I realized how interested I truly am in the concept of machine learning. While I have previously taken an Introduction to AI class, I feel like I did not get as much hands-on training as I hoped to in regard to models. And, as I am taking a separate machine learning course in the Spring, I figured I should delve deeper into this interesting topic.

I wanted to make a chatbot next, plain and simple. As I wondered which character or voice to use, an idea dawned on me.

I'm a creative, and I've made at least around 200 creations in my lifetime, counting stories and characters. A project I'm in the early stages of right now is a database of every creation I've made, to be hosted on a website using Flask. While the technical architecture of the website is outside the scope of this report, I figured it may be hard for users to navigate. The best solution to this problem?

Make one of my characters into a chatbot that the user could talk to, whether for help navigating the website or just in general.

I thought about one character in particular. Back in 2020's quarantine, I wrote a book that spanned over 200 pages. While the book was never finished (in fact, it was only around halfway complete), it holds more than enough material to train a model on.

In this report, I will first explain a bit about the book and the character. I will then go into the tools that helped me make him into a chatbot. After that, I will explain why I thought fine-tuning was a good idea at first, but completely abandoned the idea by the end. Finally, as this chatbot still ended up in rough shape, I will describe my future work to improve it and eventually deploy it to my website.

# What is Romance in Italy?

For context, *Romance in Italy* is the name of the aforementioned book, and serves as the dataset for this project. Before I delve into the technical aspects of this project, I will provide an overview of the book, including relevant information that may affect the model.

*Romance in Italy* is the story of Toby Plaus, a freshman in high school. The book's central conflict is a personal one; Toby must move on from a falling out with his old friend Bella. Throughout the book, Toby has strange dreams in which he rediscovers old parts of his childhood and reflects on his memories. The book also focuses on the new kid, Ed, and the interpersonal conflict between him and Toby as he becomes closer with Bella. Additionally, *Romance in Italy* focuses on Toby's friend group, consisting of Luke, Peter, and Dean. As the book progresses, Toby's old friend, Caleb, eventually passes away and soon begins to inhabit Toby's dreams. Alongside Caleb, Toby also begins to see a mysterious red eye in his dreams, and Peter begins to dream about the man that broke his family apart. The friend group goes back to Faith Creek, a stream where they once built wooden structures, to rebuild in honor of Caleb. This information is what I'd consider "critical" and is key for the model to know, as it is at the book's core. Thankfully, the final model had most of this down perfectly, although I will explain why later on.

Toby, as a character, is a bit anxious, a bit artsy, and a bit romantic — all aspects that I wanted the model to capture. Near the start of the book, for example, he claims he has a crush on another girl as a way to move on from Bella. However, as *Romance in Italy* progresses, Toby begins to grow. By the end of the book, the major topics have shifted, and Toby is thinking about Caleb and his friend group more than Bella. One could argue that *Romance in Italy* suffers from issues with its cohesion, giving the changing stakes throughout the book. While I agree, this level of variety also provides Toby with far more content than if the book remained focused solely on high school drama. However, it also provides a catch — as the tone changes, a model's tone may change or become confused. Additionally, the out-of-order dreams provide the challenge of piecing together a realistic timeline, and for the model not to place events randomly.

The corpus of *Romance in Italy* is rough in retrospect; however, given its vast amount of content and unique writing style, Toby Plaus was the clear character to choose for this project. The sheer amount of training data from Toby's perspective alone would be more than enough for any model to learn his voice, and adding the other characters as well would provide additional context. However, as seen below, the different perspectives would also confuse the models.

# Technical Stack

Initially, I planned on fine-tuning a Meta Llama-3.1-8B-Instruct model. To do this, I used Google Colab. Early on in the project, I utilized HuggingFace to load the Llama model and to save the resulting Toby models. Additionally, I used the Unsloth library for faster training and saving. As I trained, I realized I needed more VRAM than Colab's T4 GPU could give me; for this reason, I put $10 into a RunPod environment and continued on an NVIDIA GeForce RTX 3090 GPU. For reasons I will explain below, I started with a mix of fine-tuning and Retrieval-Augmented Generation (RAG), the latter of which essentially allowed the given model to search through a text in the present moment. Afterwards, I switched to just fine-tuning. Finally, I switched back to RAG and stuck with it. For the latter approach, I chose to access the Llama model via Groq, as it allowed for faster processing. I utilized the Facebook AI Similarity Search (FAISS) library for a vector search, while I used LangChain to manage the RAG pipeline, connecting the model prompt to the document retrieval. Finally, I used Streamlit to deploy the app.

# Some Questions for Toby

Before I touched the model, I felt like I needed to give it more information to understand Toby. Because of this, I created 98 scenarios for the model to learn from. They each followed a format of {instruction}{context}{response}. Essentially, the {instruction} string was a prompt given by a user. The {context} string described how Toby would feel after hearing it, and the {response} string was Toby's answer to the query. These handcrafted examples were not meant to be taken as fact, but rather for the model to learn Toby's voice from. Considering that I am the one who authored Toby, these would only help the model feel even more authentic. Although I initially planned to have more, I figured that this many prompts would more than cover Toby's voice, as he is generally a consistent character and has clear patterns to his behavior.

# Fine-Tuning: The Finn Problem

I only trained the first model on the questions, as I was initially unsure if training it on *Romance in Italy* would be a good idea; given the book's vast amount of content, I feared it would confuse the model. However, I ended up using RAG to give the model the book. I trained this version of the model with a learning rate of 5e-5 (as I wanted to keep it small for the small dataset) across 300 steps. This version of Toby was a disaster. Although the questions gave the model his voice, it utterly failed to remember details from the book and hallucinated many details. For example, I asked it about "Hartrov", the name of a song he composed in the book. Its response? It referred to it as a city he had once invented, and compared it to another made-up city never mentioned in the book. The loss value dipped below 0.05 in this case, leading to a strong fit with the questions. However, Toby clearly did not remember the events of *Romance in Italy*.

I was not quite sure why the RAG utilization failed so poorly in this version of the model. While it ended up working far better for the final model, the initial results scared me off from RAG for the next several days. As I will explain below, I made an effort to include more information in the next implementation of RAG.

For the next version of Toby, I trained it on the questions as before; however, I decided to train it on *Romance in Italy* as well. I set both the training rank and alpha value to 32 so the model would not look too closely at the data nor retain everything. I used an increased learning rate of 2e-4, allowing the model to learn the data better than before. Finally, I decided to keep the 300 steps, as it felt like a fine amount, and I did not want to train for too long. Although I do not have the specific training data, the loss value never fell below an average of 1.5. While I had only tested the previous model in Colab and therefore never downloaded it, I intended to so for this second approach.

Thus, the problems began.

As far as I could tell, Colab did not physically have enough resources to generate a GGUF file, which I had hoped to use in LM Studio to test the model. Therefore, I rented an RTX 3090 from RunPod to do so. In RunPod, I faced issues such as imports and file path inconsistencies, but eventually exported the GGUF from the terminal.

This one hallucinated as before. For example, I asked it who its best friends were. Its response?

"Luke and Finn."

*Romance in Italy*, if you recall, has a Luke. However, it has no Finn. The bot continued to fabricate lies, inventing a love interest for Toby named Sophia.

At this point, I had decided I was done with these easy models and went back to RunPod to train the model yet again.

# Overfitting: The Wingdings Problem

I set the rank to 128 and the alpha value to 512, hoping that the model would retain far more information. While I kept the learning rate the same, I set the model to take 3000 steps.

This model took 10 hours to train: I set it up in the morning and watched it finish around 11:00 PM. Even after it finished, I ran into other issues regarding the GGUF. For instance, my file was compressed and needed to be uncompressed before I made it into a GGUF. Then, I needed to quantize the GGUF into a smaller format for it to run more easily. Overall, I ran into several issues, which were not helped by Unsloth's long processing times. Eventually, at around 3:30 AM, I finally exported the GGUF files in the terminal and was ready to download them.

I have yet to mention a crucial detail. The loss value for this model went down to 0.007. At the time, I didn't care — I took the nuclear option, hoping for no more hallucinations. However, this was the biggest mistake I'd made yet.

At 4:00 AM, I opened LM Studio, eager for a working Toby. However, the model immediately began to quote *Romance in Italy* verbatim and generated passages similar to those within the book. In my exhaustion, I hoped for better results and simply typed "Hello Toby". However, in this moment, I had forgotten that this was a line also spoken by the mysterious eye within one of Toby's dreams, immediately followed by Wingdings. In the book, these were supposed to add to the confusing nature of the entity. However, as they were part of *Romance in Italy*, they, too, were included in the training data.

To my horror, the bot returned Wingdings gibberish and had to be stopped manually. This was quite a sight at 4:00. After this, I decided I was done for the night.

The next day, I tried merging this model with Llama in hopes that they would balance each other out. However, that, too, led to problems.

# Merging: The Repetition Problem

I set the overfit model to only 60% of its original weight when merging it with the Llama model, in hopes that the latter would dilute the former and give it some common sense back. This led to its own issues; I had problems yet again regarding the GGUF conversion, mainly due to the compression and structure of the newly merged model. The file contained compression artifacts that were immediately detected and blocked upon trying to convert its uncompressed version to a GGUF; however, by removing these, the model's size and structure were modified, making it unopenable. Eventually, I fixed these problems and opened the GGUF; however, this one was no better. It constantly repeated things such as its system instructions and other parts of the book, making it unusable. While it lacked the Wingdings of the previous model, it was no Toby: it was a monster.

At this point, I felt defeated. I had spent several days and $10 training the model to sound like my character, but nothing was working. However, after thinking a bit more, I decided to try RAG one more time.

# RAG: The Hannah Problem

At this point, I should explain what RAG actually is. As mentioned earlier, RAG stands for Retrieval-Augmented Generation, and is a way for a model to read data. RAG vectorizes a document, allowing the model to retrieve relevant information when submitting a query (“Retrieval Augmented Generation”, n.d.). Additionally, it can improve prompt handling compared to simple pretrained models and retrieve data in real time (“Retrieval Augmented Generation”, n.d.). As one last effort, I fully invested my energy into this approach. Rather than fine-tuning the model at all, I decided to utilize RAG for each part, including the questions and the book. Additionally, I created a “Toby Bible,” which was essentially a document containing all of the important facts in the book, including the “core” information mentioned above in the “What is Romance in Italy?” section. I attempted this not to “hardcode” the model’s responses or logic, but rather to guide it in the right direction. This model showed to be the best one yet, yet still suffered some issues. For example, Toby incorrectly remembered some information. One instance of this is regarding characters. Luke’s girlfriend’s name is Hannah, and she and Toby practically never talked in the book; however, the model repeatedly claimed to know Hannah as a close friend. As a whole, however, the model still felt the most like Toby. Due to this, I decided to deploy it to Streamlit. While the latter three of the four original models are now hosted on Hugging Face, the newest one is not, simply because it’s a regular Llama model using RAG. While the model is done, there is surely more work to be completed before I feel comfortable integrating it into the creations website.

# Future Work

Although I feel like I've made several modifications to the "Toby Bible" by this point, the model continuously fails in certain situations. Another catastrophic slip-up was when it spoke about Toby in third person as a different character; if it was speaking about him, I have no idea who it was trying to imitate. I suppose that fixing this model involves tightening up the Bible the best I can and understanding the best ways to format a text for a model to make the most sense out of it. While the questions were initially formatted the way they were because they were being trained upon, I may change their layout to make them easier for the model to understand. Additionally, once the model is done, I will need to learn how to deploy it with Flask to put it on my website. Overall, I have a lot of work to do; however, by setting realistic goals and mapping out the issues with the model, I can work to make it as close to Toby as possible.

# Conclusion

This model is clearly far from perfect. However, I believe I have learned a lot in the past week. Between learning about Unsloth, comprehending the GGUF format, using RunPod for the first time, implementing RAG, and truly understanding the horrors of overfitting, this project has taught me a substantial amount. While there is still work to be done on the model, I believe I can reasonably improve it. While hallucinations are inevitable, perfecting the "Toby Bible" and other documents will allow this model to truly encapsulate Toby's personality and memories.

The text of *Romance in Italy* will be included in this GitHub repo, and the link to it can be found in its citation. This cleaned copy removes chapter headers and reformats the changes in point of view to make them easier for the model to understand. At this time, I do not feel comfortable sharing the original Google Doc for the book, but its entire contents are within the aforementioned text file. While *Romance in Italy* is far from perfect, especially nearly six years later, it served as a great dataset to provide the models with.

There is more work to be done regarding machine learning, and I am excited to attempt more experiments in the future. In the meantime, give the Streamlit model a try! Ask Toby about Caleb or Luke — I'm sure he'd love to tell you all about them.

# Citations

Brassil, J. (2020). *Romance in Italy*. https://github.com/SirCacto/TobyPlausBot/blob/main/Library/Romance_In_Italy.txt

Chase, H. (2022). *LangChain* [Python]. https://github.com/langchain-ai/langchain (Original work published 2022)

*Facebookresearch/faiss*. (2026). [C++]. Meta Research. https://github.com/facebookresearch/faiss (Original work published 2017)

*Meta-llama/Llama-3.1-8B-Instruct · Hugging Face*. (2024, December 6). https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

*Retrieval Augmented Generation*. (n.d.). Databricks. Retrieved January 8, 2026, from https://www.databricks.com/glossary/retrieval-augmented-generation-rag

*Streamlit/streamlit*. (2026). [Python]. Streamlit. https://github.com/streamlit/streamlit (Original work published 2019)

*Unslothai/unsloth*. (2026). [Python]. Unsloth AI. https://github.com/unslothai/unsloth (Original work published 2023)