Table of Contents

Origin

When I was 12 or 14, an elderly kid in the neighborhood told me about Apple, Mac and Steve Jobs. I genuinely thought he was bullshitting me, making up stories and thinks I do not know anything about the outside world. Little did he knew my school had a computer lab and I knew DOS and Wordstar and I knew there was only one person who mattered and that was Bill Gates. For context, I grew up in a small town and even in year 2000, across the entire town, we only had one internet center and it had only one desktop PC. Yes, in those days, there were a thing called browsing centers and it usually had around 10 desktop computers and you must pay per hour to access the internet.

Fast forward to 2007, I was in Texas visiting US for the first time. I was looking to buy gifts for folks back home. I was looking at MP3 players for my cousin. The friend who was helping me was like, “Dude, if you really want to give a good one, get the Apple one, if not it doesn't matter which one you buy.” Again, Sony Walkman was an aspirational thing for me and someone telling me not to buy Sony Music player but an Apple one was intriguing. Not to mention the price difference. The iPod mini was too expensive when compared to other MP3 players and unlike other players, you cannot just copy songs to them like you would do in a USB machine. The usual inquisitive (a.k.a cynical) me was thinking, “is my friend recommending this because they think, higher priced means usually the best or do they really know what they are talking” They walked me through the store and showed me Apple products and was like, “Have you never heard of Apple?”

The push and the sermon from my friend to make me buy an iPod mini sent me across the rabbit hole and read anything and everything about Steve Jobs. I was so fascinated and became a hard core convert. I was so hard core that during my next trip, when a friend asked me to buy a MacBook for him, I was like, why are you buying a low-end model?.The said low-end model was some 1200+ USD and though USD INR was at 38, 1200+USD was pretty big deal even then. Oh, and he was not even an engineer but a fashion designer. (Yes, that friend also wanted me to buy a Diesel trouser and converse shoes and more colleagues in US came to see the 200USD trouser than the mac.) The trip after that I bought an iPod touch as I couldn’t afford to have an iPhone.

Since the time, I read all things about Steve Jobs, I also had a strong urge to write a book on him in Tamil. More like for the hundreds of Kids like me in hinterlands who never knew about Steve Jobs.

I wrote my first chapter of the book in May 2009, third chapter by October 2009 and I never made any progress. By May 2011, I started writing a weekly column in an internet magazine and by October wrapped up the writing, just around the time Steve Jobs passed away.

If one had studied Jobs very thoroughly, one would understand and appreciate that portrayal of him by the mainstream media is far from true. Whenever I saw any reference to jobs in the articles covering the toxic traits of founders and tech bros, my blood truly boiled. I think that has frustrated me as though someone wrote something very wrong about me.

So far long time, one of the other strong urge and desire was to write two books on Steve Jobs

One, compile all his quotes and put it as a book.

Two, based on quotes and happenings, write about how one could learn from Jobs to make a dent in the Universe rather than just being a painful "A" and fake projecting that you are channeling your inner jobsian aura.

I started compiling all his videos, interviews, books that I missed earlier, and the books published after my book. (I had to rely a lot on iCon and lot of early articles across publications like Forbes, BusinessWeek, Time, Wired and Fortune to write my book; The Walter Isaacson wasn't published when I was done with mine)

However, I never got around to writing them. Though I created the playlist and was listening to them regularly, I couldn't even get around to getting the transcripts from the YouTube video complilations.

Later, with the advent of ChatGPT, NotebookLM, I started to tinker around writing scripts to compile the transcripts from videos. At the same time, I also wanted to be AI literate and was reading around and playing around with the AI tools.

I also thought rather than reading each and every line in the book, identifying quotes and copy pasting them, classifying them based on themes, format them and doing all of it manually, why not use AI to do all the above. Thinking that way was more easier than actually making it happen.

More I tinkered around, I ended up finally across the customGPT. Ended up thinking, wow, I should have this custom GPT and make it part of the book. It would be a good branding and marketing for this book.

When I read more about customGPT (by reading I mean chatGPTing and going through the responses, I felt, building a customGPT would be easy

Crafting the GPT

Inside the Machine

What happens under the hood

Building the mind of Jobs

Lessons in Structure

20 Files limit

The 20-file limit per Custom GPT exists mainly due to a combination of technical, usability, and performance-related reasons. Here’s a comprehensive and unbiased explanation:

🔧 1. Performance and Retrieval Efficiency

Custom GPTs don’t ingest the entire file corpus into memory. Instead, they use retrieval-based methods:

- When a user asks a question, the system searches across uploaded files for the most relevant chunks using embedding-based retrieval.
- If you upload too many files, the retrieval latency increases, and the chance of pulling irrelevant or conflicting context goes up.

🧠 Why 20?
Empirical testing likely showed that performance (speed + accuracy) begins to degrade beyond 20 files per GPT.

📚 2. Encouraging Structured Knowledge

Allowing unlimited uploads might:

- Encourage data dumping (e.g., uploading 50 books without curation)
- Reduce the model’s effectiveness because it pulls too much or irrelevant material

The 20-file limit encourages users to:

- Curate, summarize, and structure their knowledge
- Upload only what’s thematically relevant
- Avoid turning the GPT into a noisy, unfocused assistant

🖥️ 3. Interface and Memory Constraints

On the backend:

- Each uploaded file is embedded, indexed, and stored
- The search process scales linearly with the number of documents
- OpenAI optimizes for interactive use, not building a full research database or retrieval-augmented LLM at massive scale

The UI also becomes unwieldy with more than 20 files — editing, updating, or understanding file interactions becomes confusing.

⚖️ 4. Resource and Cost Efficiency (for OpenAI)

Every file uploaded:

- Is tokenized, embedded, stored, and retrieved at runtime
- Incurs API-level costs and storage (especially with high-usage GPTs)

Limiting file count:

- Controls compute and memory usage
- Keeps things efficient for both OpenAI and end-users
- Avoids overuse of infrastructure in GPT Builder compared to API-based solutions (like LangChain or Pinecone)

🔐 5. Privacy and Security Management

Smaller file sets:

- Are easier to audit and manage
- Reduce attack surfaces for prompt injection, data leakage, or file misinterpretation

OpenAI must balance flexibility vs. control in a general-use tool like GPT Builder.

🧭 What You Can Do Instead

If 20 files isn't enough, here's what works:

Strategy	**Benefit**
Merge related files	Group thematically (e.g., combine all "Product Design" notes)
Rotate files	Swap files in and out as needed (e.g., leadership week vs. design week)
Summarize long files	Use GPT itself to summarize chapters or talks
Use LangChain + Vector DB	No file limit, full control, scalable retrieval from thousands of docs

✅ Summary

Reason	**Explanation**
🔄 Retrieval Efficiency	Keeps searches fast and relevant
🧠 Encourages Curation	Forces cleaner, topic-based uploads
💻 Backend Limits	Embedding and indexing scale poorly with 50+ files
💰 Cost Control	Reduces compute/storage overhead for OpenAI
🔐 Privacy/Security	Limits attack vectors and mishandling

2 million Token limit per file

What do you mean by there is a 2 Million Token limit per file? And first what is a token?

This is a **core architectural limit** in how the **ChatGPT Custom GPT Builder** processes and "understands" your uploaded knowledge files (the ones you attach in the *Knowledge* section).
Let's unpack what that **"2 million tokens per file"** actually means, and why it matters for your GPT's performance, memory, and reasoning depth:

1. What a "token" is

A **token** is a chunk of text (a few characters or a short word).
Roughly speaking:

- 1 token ≈ 4 characters of English text
- 2M (two million) tokens ≈ about **1.4–1.6 million words**, or **2,500–3,000 pages of text**
  So when the GPT Builder says *"2 million tokens per file"*, it's referring to the **maximum amount of text content** the system can process and index from any single uploaded file.

2. What happens when you upload a file

When you upload a file (PDF, DOCX, TXT, etc.) into your Custom GPT's *Knowledge* section:

- The system **ingests** it by converting the text into tokens.
- Those tokens are then **embedded** — meaning they're transformed into vector representations that the GPT can later *retrieve* and use to answer queries.
- The embedded data is **not memorized verbatim**, but stored in a **semantic retrieval layer** — a vector database that allows the GPT to "recall" relevant chunks when prompted.

Each file is processed independently, and **the 2M token cap applies per file** (not across the entire knowledge base).

📚 3. Why the 2 million token limit exists

It's primarily a **technical and performance constraint** designed to:

- Keep **embedding computation and storage efficient**
- Ensure **retrieval speed** stays under real-time limits
- Maintain **semantic accuracy** (too large files reduce retrieval precision because the index gets diluted)

If you exceed this limit, the system will **truncate** the file or reject it, depending on the upload method.

💡 4. Implications for builders

- If you have a very large corpus (say, a full book series, or hundreds of pages of documentation), it's better to **split it into smaller themed files** (e.g., *"Jack Ma Quotes – Book 1"*, *"Jack Ma Quotes – Book 2"*).
- Keep each file **focused on one conceptual domain** — this improves retrieval accuracy dramatically.
- The GPT will **search across all uploaded files**, but retrieval is more reliable when content isn't lumped together into a monolith.

🧠 5. How it impacts GPT reasoning

- Files aren't *memorized* — they are *retrieved* on demand.
- When you ask a question, the model searches its vector index for semantically relevant chunks (usually 500–1,000 tokens per retrieved segment).
- So even if your file has 2 million tokens, the GPT only "reads" the **relevant few thousand tokens** each time.
- This means: large context = broader coverage, but not necessarily deeper reasoning unless well-structured.

✅ 6. Best practices

Goal	**Strategy**
High accuracy	Break knowledge into 100k–500k token thematic chunks
Fast response time	Avoid maxing out token limit; retrieval time grows with size
Easy updates	Upload new versions as separate files rather than overwriting one massive file
Transparent retrieval	Name files clearly — the GPT uses filenames in its internal indexing metadata

⚖️ 7. Overall significance

The 2M token cap defines the **maximum semantic “chunk” size** the model can efficiently *learn from* at once.

Think of it as the **file-level comprehension boundary** — within which your GPT can form a reliable, retrievable mental map of the content.

If you’d like, I can show you how to **plan the ideal file structure and token distribution** (e.g., how many files and how to break them down by theme or topic) for your custom GPT — say, for your *WWJD?* GPT or your *Jack Ma Quotes*project — so you maximize both retrieval precision and interpretability. Would you like me to do that?

Beyond GPT

Reflections