Digital Personal Assistants

published on March 14, 2023

Large language model (LLM) technology—in which a lot of computation makes a system that can autocomplete text uncannily well—has recently advanced by leaps and bounds. (It's commonly called "AI," but I'll avoid that term.)[1]

I was first impressed by the field's progress in the summer of 2020, when OpenAI released an API for "GPT-3", the first publicly-available LLM that could generate passable writing. It made waves in the tech community, with people prodding it to make webpage components, news articles, Dr. Seuss-style poems, and more. I remember being especially blown away by a website could simulate conversations with a (frequently misguided) expert on any topic you could imagine.

But nothing really came of it at the time, and the wave of hype died down. Nobody outside of extremely-online tech circles ever even heard about it. A few "AI" companies limped into existence by trying to browbeat the GPT-3 API into doing anything consistently well (a very difficult task!), but most people moved on.

Nonetheless, OpenAI kept at it. They released GitHub Copilot, a decent multi-line autocomplete tool for programming languages. They kept shipping new and better variations of GPT-3.[2] They made a splash by releasing DALL-E 2, an image-generation model.[3] And finally, after months of frustration that other people weren't using their API to its potential,[4] they built and released a basic app called "ChatGPT" to the Internet for anyone to use.

Everyone knows what happened next, because almost everyone reading this has used it: ChatGPT *exploded* in popularity. Two months after launch, it had *100 million* users; that's faster growth than *any product in history!*[5] Microsoft, smelling blood in the water, quickly added LLM-based features to its unused search engine Bing,[6] then gloated at their newfound ability to “make [Google, ‘the 800-pound gorilla in search’] dance.”[7] And they weren’t exaggerating: Google, in a panic, declared “code red” and paged its founders for the first time this decade to plan a response.[8]

ChatGPT may have been the first big LLM-based product, but it certainly won’t be the last. What will be the most impactful one be? Is there any progress to make? The far future is unknowable, but I am confident that at least one big thing—a product I call the “digital personal assistant” (DPA)—is on the horizon.

Today, a few fortunate people have personal assistants. These humans follow them around, listen to them talk, and are generally very familiar—from long exposure and study—with their boss’s thought patterns and preferences. Personal assistants can save a *lot* of time; they handle logistics, assemble and keep track of schedules, flesh out rants into eloquent speeches or scribbles into publishable articles, filter inputs to their master’s mind (like email or letters), and more.

But typically, it’s only only the highest-leverage information workers—CEOs, elected politicians, and so on—that have them. That’s because all personal assistants are *people!* To shave off a few hours of boring tasks here and there, you have to find a generally well-educated and smart person, convince them to help you, move them to your (presumably expensive) area, and pay them a full-time salary. No part of that is cheap… but at a certain point, your time is worth the price.

The development of *digital* personal assistants will reduce that price substantially, saving everyone time by democratizing access to them. You, I, and the dumbest person you know will soon have nearly-free software agents, intimately familiar with the state of each of our minds and desires, that can eliminate most mundane digital drudgeries from our lives.

Modern LLM-based products are getting increasingly close to this product. But even New Bing, the best I’ve used, is still missing a few key features. I’ve identified five upgrades that would add up to a digital personal assistant, and that are all readily implementable with today’s technology:

1. the ability to use arbitrary tools
2. the ability to work through arbitrary interfaces
3. a persistent memory, which would lead to a consistent identity
4. the ability to ingest external personal context about their users
5. the ability to operate independently and automatically

The net result of these additions, I hope to show, will be a functional DPA. It may still require better LLM models to be developed to be *good*;[9] but *something* like this, I am confident, is possible. A proto-DPA with the basic architecture could be built today.

And consistent architectures are powerful, because they allow for component upgrades.[10] As improved LLMs become available, you’ll be able to swap them into your DPA as you’d put a new CPU into a desktop computer; and as new and better tools or interfaces get developed, you can plug them into your DPA, just like a new screen or keyboard for that same computer.

1. tool usage

Pure-LLM products’ biggest flaw is that they fluently hallucinate information, asserting falsehoods and inventing fake statistics or citations like there’s no tomorrow. This behavior stems from the fact that LLMs are optimized to make text that *looks correct*, whether or not it is completely made up. Multiple “AI search engine” products[11] solve this with a one-two punch: by (1) asking LLMs not to “remember” information themselves, but instead only to *summarize* external sources,[12] and (2) by giving them control over a search engine,[13] so that they can go and find sources to summarize for any topic you ask them about.

This approach can be extended, and surely will be soon: if an can LLM use[14] a search engine, why not other tools?[15] Logical candidates might include:

- the ability to execute arbitrary computer code that it writes
- the ability to ask science/math questions of Wolfram Alpha
- the ability to make any HTTP request
- the ability to communicate with humans (via email/phone/Twitter/etc)

A key feature of digital personal assistants will be the ability to use not only these but *arbitrary* tools in this way, in a “plug-and-play” manner. You’ll be able to code up (or download) a “search engine tool”,[16] or a “calendar tool”, or an “algebra solver” tool, or a “send email” tool… then “plug them into” the DPA,[17] and let its core LLM use them at its leisure.

2. interchangeable interfaces

Ever used Siri? Alexa? “Okay” Google? Yeah… they suck.

I once heard a story about an executive at Amazon that got an Alexa before they were publicly available and found it *super* convenient for setting timers hands-free while cooking. “If it’s this helpful in the kitchen,” she imagined, “how much potential power is just *waiting* to be implemented?!” So she moved over to the Alexa division… only to discover, after years of disheartening development, that setting timers was basically the only thing Alexa could do well.

I get it. I have HomePod minis littered all throughout my house, and have found that they’re useful for precisely four things: playing music, turning lights on/off, setting timers/alarms, and doing basic arithmetic.

The reason “smart speakers” and “voice assistants” suck is that they’re fundamentally textual: you say words at them, they say words back to you. But most of our technology is visuospatial: we work with GUIs, look at pictures, make sense of things through their position and color, and take action by tapping screens or clicking cursors. For an assistant you can only talk with to be helpful, they have to be *really good* at language; and previous NLP (“natural language processing”) technology has been, well, really *not* good at language.

But LLMs change this. If a DPA can be helpful over a web interface—like ChatGPT or New Bing—where you type in text and get text out, it can be *just as helpful* over a voice interface![18] Instead of banging your head against Alexa or Siri, you'll soon have a DPA accessible by just opening your mouth.

DPAs will work with *any* textual interface; you'll be able to switch formats effortlessly, even using multiple with the same assistant. You could ask your DPA questions over iMessage, or bark commands at it like Alexa, or give it a phone call, or talk to it via the microphones in your AirPods when you press a button on your watch. Every textual interface you can imagine—and many we can't yet—will be used to access them. And this flexibility will be important: a DPA in your ear is much more useful than one you have to open up a computer and go to a certain website to talk to.

3. persistent memory

Right now, every time you talk to an LLM product, it has no clue who you are, and no memory of past interactions. This is made explicit, as if it's a feature: whenever you start talking to New Bing, for example, it proudly announces that it has "cleared the slate for a fresh start."

But I think this is a bug, not a feature. If a DPA had a memory—especially of its interactions *with you*—it would be better able to predict what you're looking for; be able to tailor explanations to your personal knowledge strengths and gaps; and so on. You would have a more persistent and productive relationship with the tool.

Just as with strong human relationships, a lot of things that you would otherwise have to explicitly tell your DPA in every new conversation would eventually be left unsaid. You would be able, for instance, to pick up where you left off in an old investigation; or analyze what you said about some person or topic in a conversation months ago. Over time, you and the DPA will gradually develop shared shortcuts and slang; and as you hone your shared dialect, you'll be able to communicate meaning more precisely to it than to than with other people or machines.

I suspect that all this could be accomplished by simply recording all past conversations you had with the DPA and giving it a “tool” (see upgrade #1) to search through or look them up.[19]

4. external personal context

Memory is great and all; but even with it, your DPA will only know what you tell it (and it can find on the Internet about you). Instead of having to tediously repeat information about yourself, why not give it direct access to your personal data? In theory, the more accurately it can understand your patterns of action and thought and preference, the more helpful it will be.

(Is your brain, at this point, screaming “what about privacy? privacy Privacy PRIVACY!!!“? Good! Mine is too. Hang on, we’ll get to that topic at the end 😊)

This could be done similarly to upgrade #4; you’d give the model a “tool” to access data about you (past location, past purchases, writing, work history, files on your computer, message history… as much or as little as you’re comfortable with, really).

Doing this would also turn it into a supercharged version of your “normal” memory. Ask your DPA “uh, when was it, again, that I bought this pen I like?” and it could go through your email receipts, location history, computer logs, and message history to tell you (1) where you bought it, (2) at what precise minute—even years ago—you bought it, (3) how much it cost, (4) what you were thinking about at the time, (5) who recommended it to you and in what way… and so on.

Having not only an assistant that knows literally everything about you, but a completely superhuman memory, will be incredibly useful. So useful, in fact, that I believe many people will pivot 180° on their fear of data collection and opt to record and feed as many aspects of their life as possible into their DPA, such as:

- live biometric data (through a smart watch)

- all text ever shown on your computer screen (some startups already do this)
- your 24/7 surrounding audio (by wearing a concealed microphone)
- the auto-transcribed text of years of your old handwritten journals

Back to privacy: this is a **lot of information** to give some company, especially if they also then have control over an assistant that has complete memory of your life and that you deeply depend on. For this reason (and a few others I'll go into later), I expect that the best DPAs will be open-source so that their users can customize them, avoid extortion (how would it feel if a company could charge you for access to your spouse? Doing so for a DPA would not be too different), and control their own data. I hope.

5. independent operation

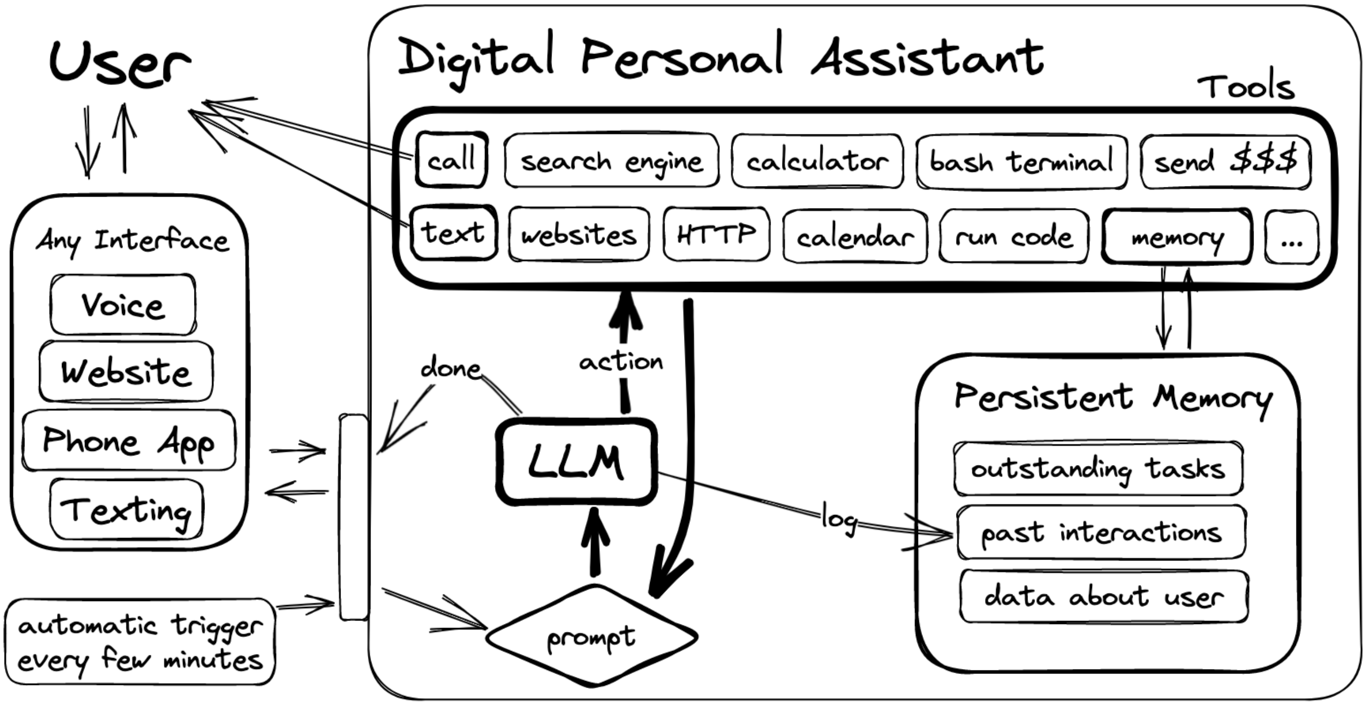
Current models operate like search engines or calculators: they're useful tools that are there when you need them, but only then; you always have to initiate the interaction. I don't think it will be like that in the future; rather, once models have a persistent "memory" and identity, they can run in the background automatically and accomplish tasks or alert you of various things you might care about.

For instance: you could tell a DPA to block any non-urgent email notifications until a certain time of day (say, 9a the next day) so that you don't get distracted… but to let any that are important or that it thinks you'd care about and want to see early (based on its knowledge of your state of mind) through immediately. You could tell it to make plans with a old friend for lunch next week, and have it let you know once all the back-and-forth is done and a date is settled. And so on; you get the idea.

This would be pretty technically simple too. Once a model has access to diverse tools and a memory, it can "remember" outstanding tasks or priorities. Then, it can be automatically called by a computer every 5 minutes with a prompt like "check your outstanding TODOs and take any appropriate actions." Another tool would let it send you a text message or other notification, prompting you to read something or give it clarification on instructions.

overall architecture of a digital personal assistant

All those upgrades, put together, look like this:



There's quite a bit of stuff there—nobody said it would be easy—but the problem is certainly tractable with modern technology. Even if modern LLMs don't make particularly good DPA hearts yet, the architecture, with its plugs for arbitrary interfaces and tools, would work today. ***It can be built.***

(I plan to prove this by building an open-source DPA core in my spare time over the next while. Feel free to follow my Twitter for updates, or my YouTube to watch.)

"closed" vs "open" digital personal assistants

DPAs will be powerful, and the leverage they give will be worth a lot of money. Because of this, it wouldn't surprise me if at least four well-positioned corporations (OpenAI, Microsoft, Google, and Apple) are separately trying to build something similar to this as quickly as possible. (If Apple *isn't* actively trying to turn Siri into a DPA, Steve Jobs is surely spinning in his grave fast enough to provide enough electricity for the entire Bay Area, so they should probably explore a pivot into power generation instead.)

But there are a few fundamental problems with corporate or “closed” DPAs!

- As we saw in upgrades 3 and 4, the best DPAs will be those that integrate most tightly with your life; the more a DPA can consistently read into what you’re thinking when you give it a command, the less you’ll need to specify for it to be effective at any task you give it. So, these tools will be highly personal. I, for one, do not want to send Google or Microsoft a full record of my life.
- The most effective uses of DPAs will be in business environments with proprietary data; in such cases, data often *cannot* be sent to other companies! So, non-tech companies will want on-premise DPAs, which provides a natural niche (and source of funding / legal defense) for open-source development.
- People will also come to develop psychological dependencies on their DPAs (because of our universal impulse to anthropomorphize things); it would be heart-wrenching to have your assistant’s personality changed or taken away from you without consent. To ensure this, users will want control.
- Big companies’ PR interests will also hinder tight customizability; they’ll be much less willing to let you customize or interface with DPAs in ways that a journalist could inflate to the extreme and scream about.

All these incentives will hinder the development and limit the quality of “closed” DPAs, at least compared to their “open” alternatives.

Because open-source alternatives will certainly exist. The idea of DPAs is so compelling that many people—starting with yours truly—promptly leap into the gap to ensure something exists that anyone tech-savvy can download and run (or non-tech-savvy people can easily rent from fungible providers). Some standard for interface and tool “plugs” will develop, then open-source users will be able to swap them out and customize their own DPAs to the heart’s content (or just stick with some solid defaults).

I’d bet that such open-source DPAs become near-universal in the long term, for similar reasons to Unix’ dominance among computer professionals and open-source package ecosystems’ ubiquity in software development: they will just evolve and improve faster. The free efforts of thousands or millions of people will pour into honing the details of open-source DPAs; and because LLMs are so finicky, corporations’ “closed” DPAs, even if they have individual tools or interfaces that are are better, won’t be able to refine the full package quickly enough to compete.

(At least, unless I’m missing something big. Please reach out if you think so!)

1. I once heard a quip that the main problem with “solving artificial general intelligence alignment” is that there is no clear definition for the words “solving,” “artificial,” “general,” “intelligence,” or “alignment.” I agree: instead of using vague words to describe technologies in fancy ways, we should focus on the actual tools themselves, and what problems we can use them to solve. Debating whether or not something is worthy of the label “intelligent” or not is a waste of time. ↩

2. These were largely retrained models with slight variations in input data and architecture. The main effects were to make GPT-3 (1) even better at generating text or code, (2) faster, and (3) cheaper. Nothing was fundamentally changed. ↩

3. DALL-E 2’s success was the genesis of a very interesting genre of image-generating models, both proprietary and open-source, like Midjourney, Craiyon’s “DALL-E mini” and StabilityAI’s “Stable Diffusion”. This boom was prominent enough to prompt articles full of wonder and worry like this one from the Washington Post. Image-generation models are worthy of another essay, but I won’t be touching much on them here. ↩

4. According to Sam Altman (OpenAI’s CEO), the model that OpenAI used to build the initial versions of ChatGPT had been publicly accessible, sitting around relatively unused, for about 10 months before they decided to build ChatGPT with it. This probably refers to `text-davinci-002` or `code-davinci-002`, which were released together in early 2022. ↩

5. https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/ ↩

6. https://news.microsoft.com/the-new-Bing/ ↩

7. https://www.theverge.com/23589994/microsoft-ceo-satya-nadella-bing-chatgpt-google-search-ai ↩

8. https://www.nytimes.com/2023/01/20/technology/google-chatgpt-artificial-intelligence.html ↩

9. I strongly suspect this is true, because current-day LLMs (which as of the time of writing are

OpenAI's `text-davinci-003` and `gpt-3.5-turbo`, and Meta's public-weight `LLaMA`) have context lengths that I believe to be too short for a meaningful DPA (4097, 4096,, and 2048 tokens, respectively). The "context length" is basically the size of the model's short-term memory; it cannot effectively analyze text or inputs longer than this. Each word is on average ~1.4 tokens long; for example, the draft of this essay that I'm editing as I write this was 3618 words long and mapped to 5133 tokens. So, a DPA based on a current-generation LLM would be constitutionally incapable of summarizing this essay, let alone comparing it with something else or doing any kind of transformation on it. I think that this is pitiful situation is likely analogous to kilobyte-size hard drives in the early days of computers; just as those gave way over time to fingernail-size 1TB microSD cards, I expect that future LLMs will eventually be able to operate over millions or billions of words at a time. ↩

[10] I'm intentionally skipping a step here to keep readers' attention. To be more precise: I think of this architecture more like von Neumann's computer architecture, which defines the roles of CPU, memory, instructions, and so forth. All modern computers use this system, but each *platform* has different specifications and standards, which allow for component upgrades. In a similar way, I expect there to be different DPA platforms and standards—ways of defining the "plugs" and interfaces between LLMs/tools/interfaces/memory—within each of which upgrades are possible, across which upgrades are impossible, and of which one or two will win-out long term. But all instances of these "DPA platforms" will be DPAs, just like all instances of "computer platforms" (e.g. phones, laptops, rack servers) are von Neumann computers; so almost everything using a DPA architecture will be upgradeable. ↩

[11] such as New Bing, Neeva, Metaphor, and a whole host more ↩

[12] This approach is still imperfect in practice, as LLMs' "summary" can distort or omit key arguments or claims of the sources. But I expect we'll get better at generating summaries with LLMs over time, and eventually come to rely on their usual faithfulness. ↩

[13] I don't think these models actually have access to search engines and HTTP web browsing proper yet, because their context lengths probably couldn't handle it. I haven't dug into the others, but I found evidence—poking around in the browser tools while using New Bing—that Microsoft actually has another service that does the searching/browsing and returns a list of relevant sources plus an excerpt from each. I assume that Bing then summarizes these into a readable answer for the user. ↩

[14] Can an LLM "use" tools? As with the question of their "intelligence," I think debate is not worth our time. What I refer to as an LLM "using" tools is generally an external program asking an LLM to generate some text, then when that text includes commands (e.g. `INPUT: calculate(2+2)`), running them, appending the result (e.g. `OUTPUT: 4`), and then calling the LLM again to continue. I believe that such an operation is easiest to talk about by implicitly ascribing agency to the LLM via use as a grammatical subject. ↩

[15] An experiment called "Toolformer" has already shown the promise of this approach. In it, some researchers at Meta successfully got an LLM tool teach itself how to use some tools that they hooked it up to: a factual-question answerer, a calculator, Wikipedia search, a language translator, and a calendar. ↩

16. Tool distribution and creation suggest two avenues for quick improvement of DPAs, once arbitrary tool use has been implemented: (1) tool markets, wherein people create and distribute certain tools for given DPA platforms; and (2) recursive self-improvement, wherein a DPA can write tools for itself, even “learning” to improve them by trial and error. ↩

17. What would “plugging tools in” look like? Probably just listing the ones available, and giving basic documentation on their use, along with the ability to find more. As we’ll see in example #3, with a persistent memory a DPA will be able to figure out how to use them. (Toolformer did this by “fine-tuning” the model; I’m not sure how well this will work at scale versus having look-up-able instructions.) ↩

18. All you have to do is convert sound to text (“transcription”), feed it into the DPA, then convert its resulting text back into sound (“text to speech” ⇒ “TTS”). Not only are both of these technologies well-developed, they’re being improved drastically by the same wave of tech that has made these awesome LLMs. ↩

19. As with other features, much of this functionality depends on the DPA’s core LLM having a much higher context length than any do today. ↩