

Everyday with **GPT-4**

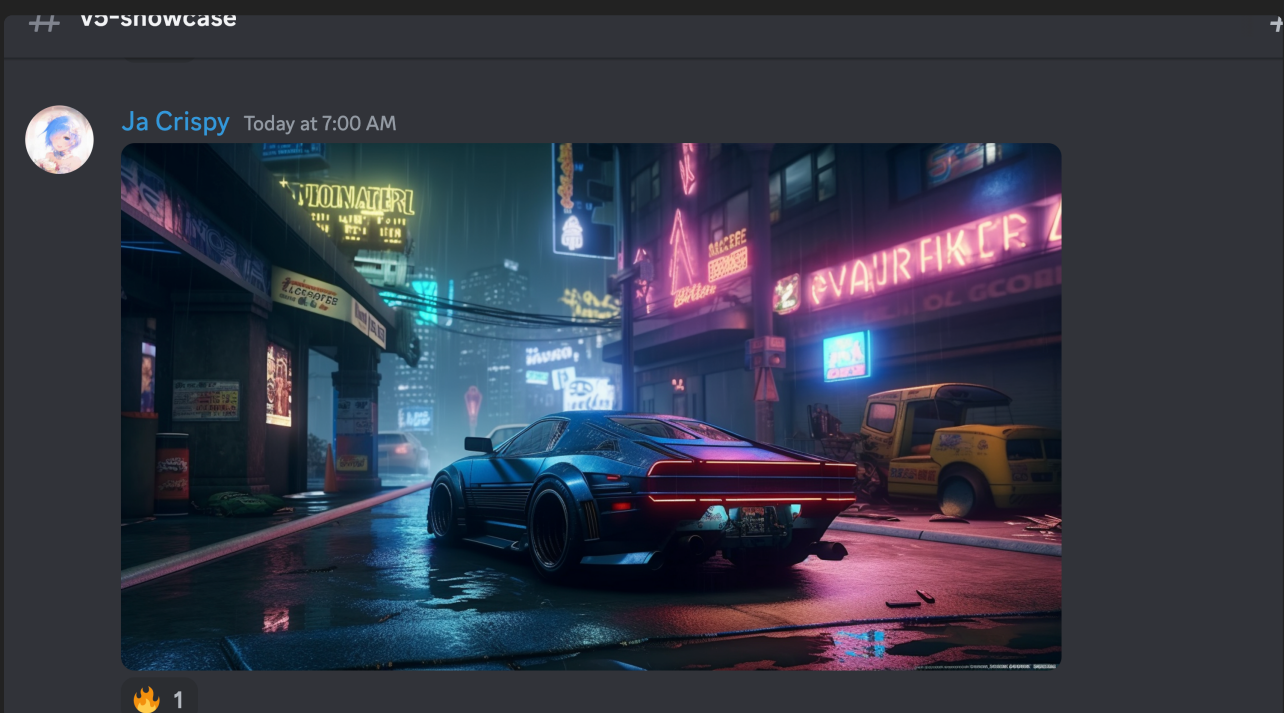
I'm not sure if at this stage we can accurately determine the impact of GPT-4 and similar tools on our daily lives. However, looking at the extraordinary images generated, such as [Midjourney](#) in version 5, and the sound quality generated by [ElevenLabs](#) based solely on a few seconds of a sample, it seems that we are dealing with something big. If we add to this the unprecedented pace of AI tool development, we only get confirmation that it is difficult to predict how our daily lives might change. I don't know if you know, but ChatGPT reached its first million users in just 5 days from the start. After 2 months, we were already talking about a number of 100,000,000.

The drawing you see below was created in less than a minute, based on a few words of my description. A few words, with the help of GPT-4, were turned into this description, and then into an image:

The girl sits in front of the 32" screen, illuminated by the neon glow of the cyberpunk cityscape outside. Her fingers fly across the keyboard as she types away in the text editor, the words appearing on the screen in a flurry of code. She is creating something new, something that will revolutionize the world.



Now you can imagine what a person who has devoted hundreds or thousands of hours of their life to learning to draw and operate graphic programs might feel. One might think that the career of someone like this is coming to an end, and like many other professions, will soon become a thing of the past. On the other hand, you can look at it differently and combine your skills with the possibilities offered by AI tools and create works like this:



I can easily imagine how games and movies will soon look, with elements created by artificial intelligence and the tone set by humans. What I'm talking about now, in the context of my work as an animator, is perfectly illustrated [by Aaron Blaise from Disney](#). Looking at it this way, it seems obvious that the best strategy we can adopt now, in the context of our professional work, is also the answer to the question: How can I combine my unique qualities with what AI offers?

This e-book is about that.

Together with Jakub "unknow" Mrugalski, we will guide you step by step through everything you need to know to start using GPT-4 in a very unusual way without any programming knowledge. Instead of just asking questions and expecting the perfect answer, I'll show you how you can very precisely indicate what you need. In addition to ready-made solutions, I will mainly share with you the principles that will take your work to a completely different level. Ultimately, just like me above, we will make GPT-4 generate commands independently and then execute them for us.

You don't have to program, but basic knowledge of Shortcuts, Airtable, or [Make.com](#) can be very useful. However, it's not essential to get the most out of this e-book that we have prepared for you.

Possibilities of GPT-4

I assume that you haven't missed the posts on social media, showcasing sometimes impressive, but very basic conversations on ChatGPT — answering questions, generating text, or writing more or less advanced code. In practice, it turns out that this is just the beginning of the possibilities. What's more, our focus won't be on ChatGPT at all, but directly on GPT-4, which we will connect to using macros and scenarios developed by me, and I will guide you through it step by step.

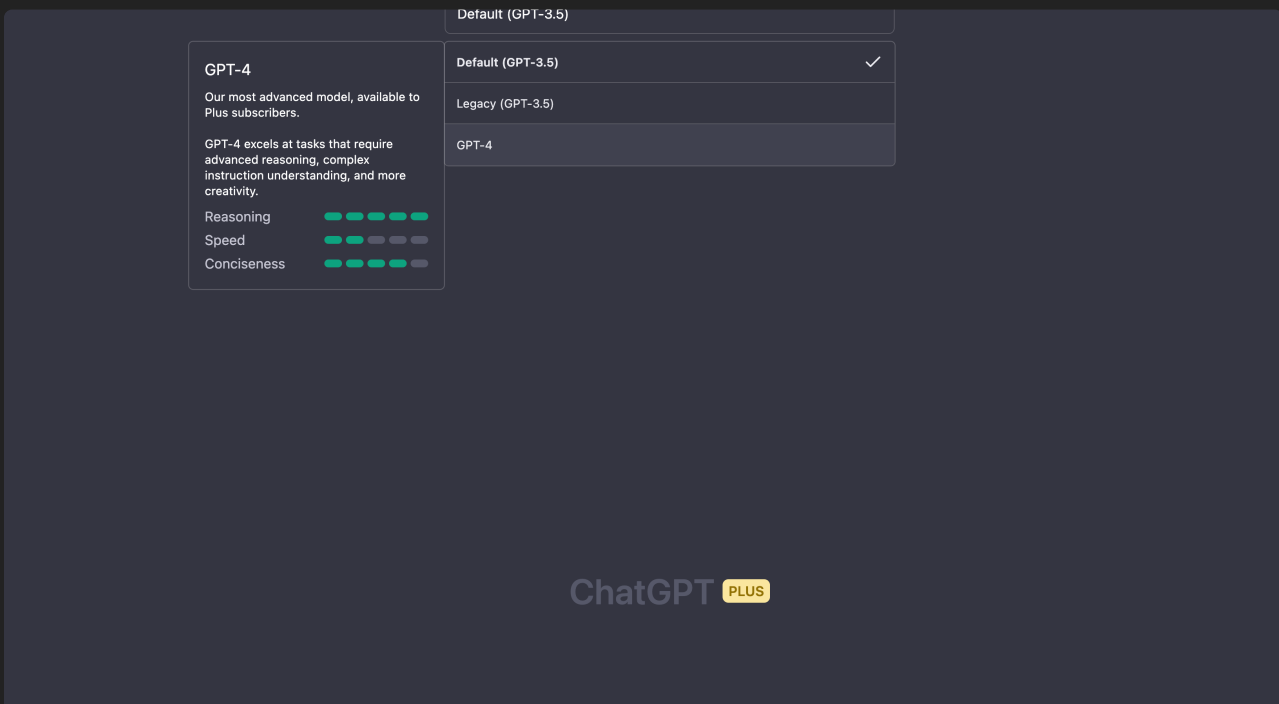
I don't want to purposely explain the details of how GPT-4 works, and I don't even see the need for it. Just imagine GPT-4 as someone very intelligent who learns quickly and has a large but limited access to knowledge. Moreover, it knows nothing about you and only a little about what you do. This means that **you can't treat GPT-4 as an oracle or someone who will do all the work for you**. At the same time, GPT-4's excellent memory and high intelligence can make it a perfect assistant for you.

I must warn you, though — GPT-4 sometimes lies or, to refer a bit more to how it works: **it doesn't always guess correctly**. And although this phenomenon occurs in GPT-4 much less frequently than in earlier versions, we still have to deal with it. For this reason, **always verify the information you receive, especially in situations where precision is crucial**. In practice, it's enough to approach the generated answers with some distance.

However, know that the effectiveness of "guessing" GPT-4 depends largely on the description of the task and the context provided (e.g., information you want to work on, containing details and instructions, helpful in performing the task).

To make it easier for you, we will present you with the best techniques based on the recommendations of OpenAI, the creators of GPT-4, as well as the experiments of other people and our own experiences.

As I write these words, it has only been a few days since the premiere of GPT-4. Access to this version of the model is available to ChatGPT Plus users, and API access is exclusively for invited individuals. This situation may change in the near future. In the meantime, I encourage you to purchase the paid version of ChatGPT and make sure that when starting a conversation, you choose the appropriate model.:



Translating this into practice, we're talking about:

- translations (e.g. Polish-English)
- summaries (e.g. bullet point summaries)
- text transformations (e.g. grammar correction)

- categorization (e.g. assigning to categories)
- searching (e.g. finding connections in text)
- text analysis (e.g. selecting keywords)
- rule-based writing (e.g. writing with specified words)
- creative work (e.g. inventing product names)
- design (e.g. generating color palettes)
- working with tools (e.g. generating Excel formulas)
- programming (e.g. generating JavaScript snippets)
- problem-solving (e.g. explaining errors)
- complex tasks (e.g. regular expressions)
- learning (e.g. generating questions and answers)
- fun (e.g. imitating speaking styles)
- conversation with AI (e.g. improving instructions)

The list is not complete, but it includes the most popular examples of using GPT-4. What is also important is that each of the above categories can be combined with each other. Our work with GPT-4 does not have to consist of issuing a single command. By combining them, we can achieve amazing results. What's more, these results will be based on the data we provide and precise instructions, the creation of which will not require writing code. I mean commands written in Polish, including individual rules that I will explain to you shortly.

And finally, probably the most important element of the whole thing is the fact that GPT-4 will be available to you anywhere on your computer and even on your phone (iPhone).

This means that you don't have to use dedicated tools and applications with built-in AI features. Everything will be available at your fingertips. We will also use the make.com platform to prepare automations that will enhance the capabilities of GPT-4 related to collaboration with various tools and applications you use. We will also make GPT-4 able to access information from the Internet, which is not available in ChatGPT.

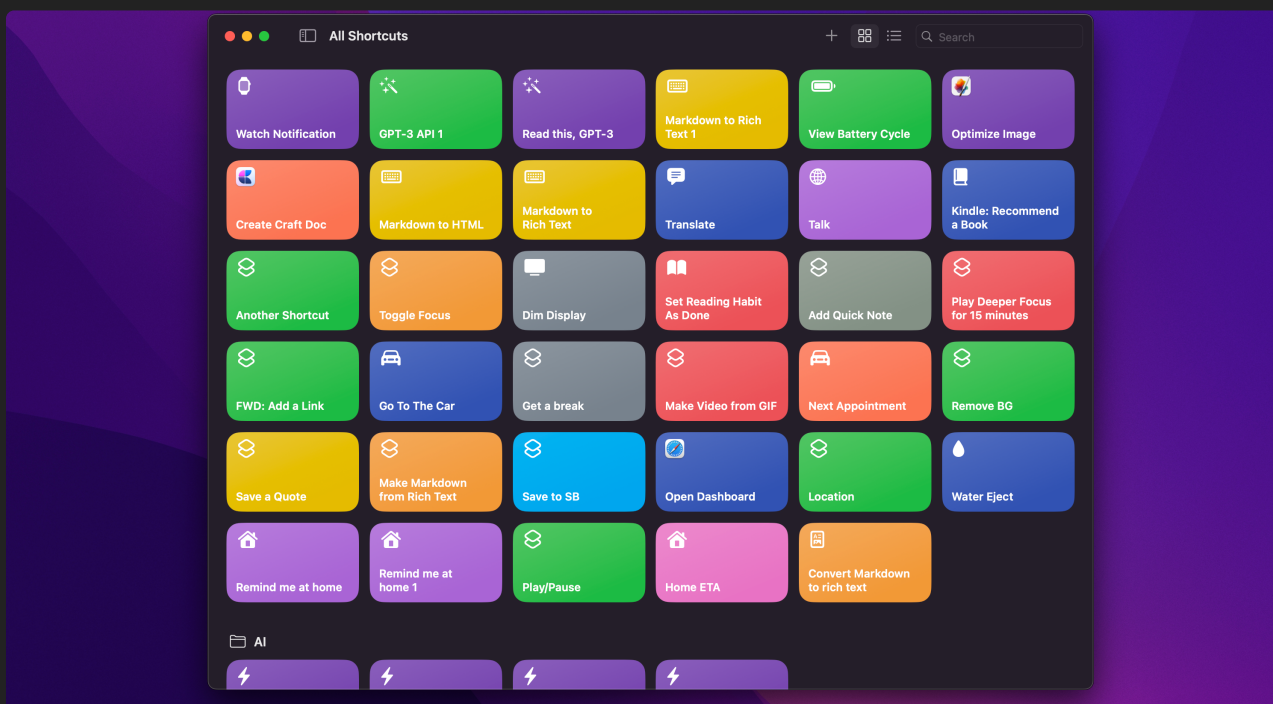
Toolset

Establishing a connection with GPT-4 requires the use of APIs and Webhooks. However, you don't need to know too much about them, as we will rely on macros, scenarios, and templates I have prepared. Your task will be to adapt them to your needs (according to my instructions and/or your creativity). I would like to quickly guide you through the tools we will be using.

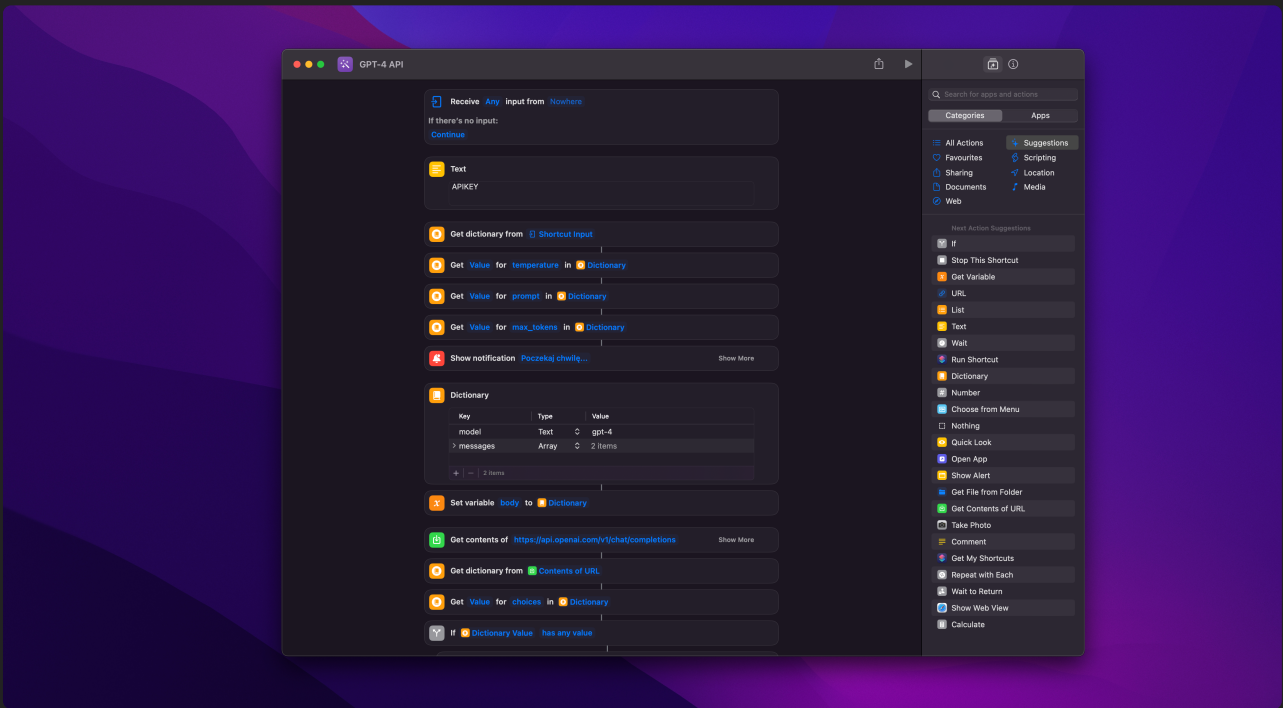
Shortcuts Thanks to the Shortcuts app available in macOS / iOS, GPT-4 actions can be assigned to keyboard shortcuts. This allows you to access them in virtually any location.

Creating and editing macros in the Shortcuts app is incredibly simple and practically requires no coding, except for individual pieces of AppleScript code (you don't need to know it). Even more so since we will be using a very small number of actions that Shortcuts offer. The rest will not be needed.

You should know that the Shortcuts app is available not only on macOS, but also on iOS. However, in practice, only selected macros can be run on the phone, which do not contain actions that work exclusively on the computer.

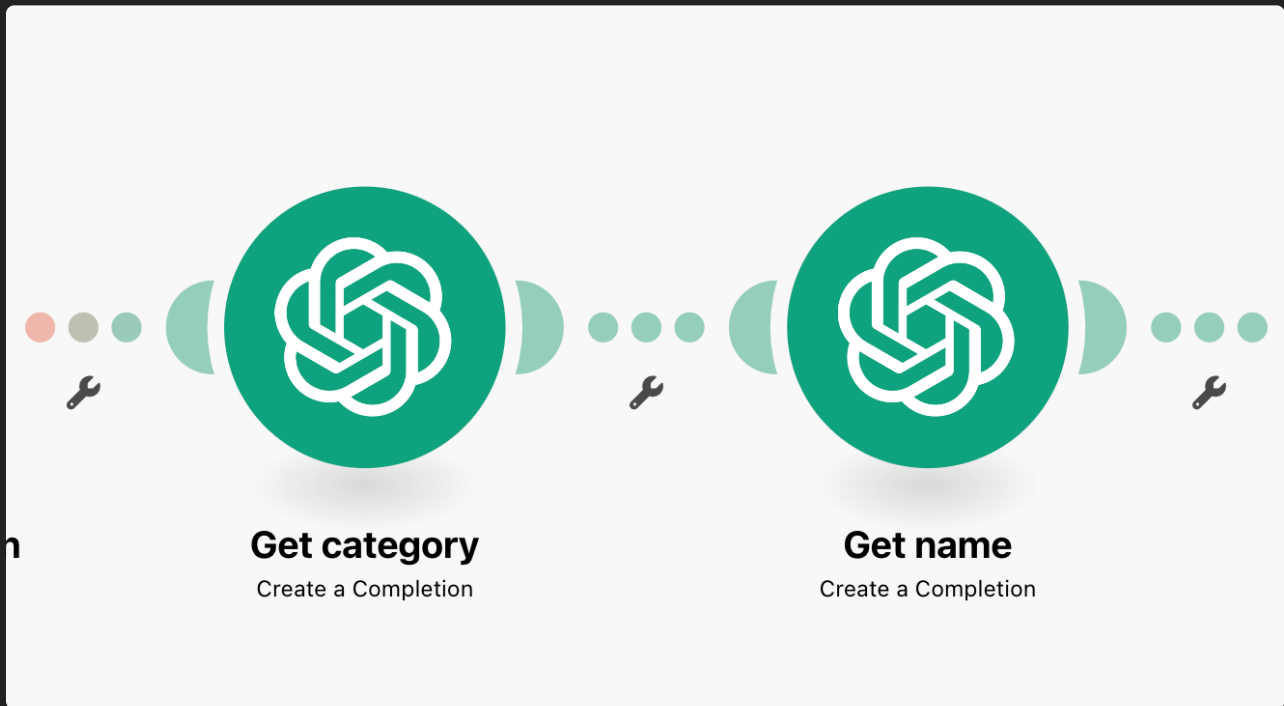


In addition, building macros in the Shortcuts app largely comes down to the "drag and drop" method. Unfortunately, I have to admit that its stability is not perfect, and more than once during the construction and testing of macros, I had to restart it because it stopped responding to my actions.



[Make.com](https://www.make.com) is a platform that allows you to connect applications without the need to write code. We will use the scenarios I have prepared once again, which you can adapt to your needs. You don't need to know [Make.com](https://www.make.com), but if you have any experience with it, it will definitely be helpful. [Make.com](https://www.make.com) is the only paid application (besides access to GPT-4) that we will be using.

Creating scenarios in [make.com](https://www.make.com) involves "clicking through" a chain of connections between modules, of which there is a huge library. One of the modules is OpenAI, and we will use it many times.



Airtable / Notion — the last set of essential tools is Airtable or Notion. We will use them as a place to store information that [make.com](https://www.make.com) scenarios will work on. Again, you don't need to know anything about them. In both cases, a free plan will be enough for you.

⚠ Important! GPT-4 Access Costs Before we proceed, you should know that using GPT-4 is not free. However, you can set spending limits on your account to avoid exceeding the limit you've set. The billing rules for access are quite complex and are based on so-called tokens, and in short, they are related to the amount of data processed and the "versions" (i.e., models) of GPT-4 that we'll be using. For everyday needs, these costs are not significant, and in my case, they amounted to ~\$5 last month. You can monitor them [here](#), and you can set limits [here](#).

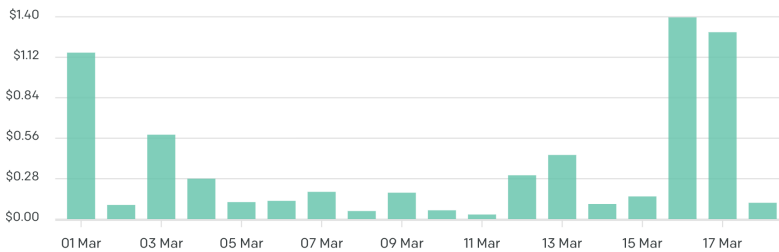
Usage

Below you'll find a summary of API usage for your organization. All dates and times are UTC-based, and data may be delayed up to 5 minutes.

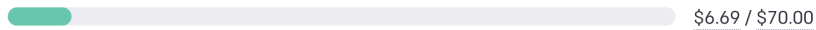
< **March** >

DAILY CUMULATIVE

Daily usage (USD) ⓘ



Usage this month



Conversation with GPT-4

Imagine now that I am here to help you with your work. I can accomplish many things for you. Despite being quite intelligent, I don't know much about your work, about you, and how you carry out your tasks. If you tell me to organize your tasks, I will probably do it, but there is a high risk that I won't do it the way you expect. However, if you explain to me how to complete the task and provide a few examples, the chances of me doing it correctly are very high.

Exactly in this way GPT-4 works, and I will show you this using the example of my task list:

Me:

Assign my tasks to categories:

- Write a newsletter
- Write an e-book
- Fix errors in newsletter automation
- Do the shopping
- Schedule a visit with the veterinarian

GPT-4:

Here are your tasks assigned to the appropriate categories:

Work:

1. Write a newsletter
2. Write an e-book
3. Fix errors in the newsletter automation

Personal matters:

4. Do the shopping
5. Schedule a visit with the veterinarian

At first glance, the result is correct, but it doesn't fit into the categories I expected: eduweb, easy_, private, others. GPT-4 doesn't know what eduweb and easy_ are, so I have to tell it about them.

Me:

Assign my tasks to one of these categories:

###

Eduweb: An online training platform where I create courses, newsletters, and a community on design & tech topics.

Easy_: A solution for selling digital products in which I do programming, marketing, and customer service.

Private: Tasks related to my personal life.

###

###

Task list:

- Write a newsletter
- Write an e-book
- Fix errors in newsletter automation
- Do the shopping
- Schedule a visit with the veterinarian

###

GPT-4:

Task categorization:

eduweb:

- Write a newsletter
- Write an e-book
- Fix errors in newsletter automation

easy_:

- (No tasks assigned to this category)

private:

- Do the shopping
- Schedule a visit with the veterinarian

The effect is better, but not only is it not what I want, but it's even far from what I need. The problem is the imprecisely given command.

Me:

Assign the tasks listed below to one of these categories. Do it in the format:

- task (category)

###

Eduweb: An online training platform where I create courses, newsletters, and a community focused on design & tech.

Easy_: A solution for selling digital products in which I program, handle marketing, and customer service.

Private: Tasks related to my personal life.

###

###

Task List:

- Write a newsletter
- Write an e-book
- Fix errors in newsletter automation
- Go shopping
- Schedule a visit with the veterinarian

###

GPT-4

- Write a newsletter (eduweb)
- Write an e-book (eduweb)
- Fix errors in the newsletter automation (eduweb)
- Do the shopping (private)
- Schedule a visit with the veterinarian (private)

Perfectly! I received exactly the result I was looking for. However, pay attention to the steps I have taken.:

- I provided a precise query, including the phrases "the list below" and "from these categories"
- I issued a clear instruction, indicating the method of recording the answer in this case
- I added descriptions of the categories, including a minimal description of each. The descriptions contain keywords that would make it easier for a person to assign tasks
- I separated the individual parts of the text with

You can now see that the entire issued instruction simply translated what needs to be done, providing the minimum necessary information to accomplish the task.

Everything you see here does not contain magical techniques, but a form I invented, sometimes referring to the suggestions of GPT-4 creators. However, I would like the rules to become clear to you, which you can use in designing your own queries. They look as follows:

- Precisely define the task
- Provide the necessary context to help answer the question
- If possible, include an example solution
- Clearly separate the instruction, context, and information to be changed
- Keep the query length as minimal as possible. The longer the query, the higher the cost (and there are length limits depending on the so-called model you're working with)

That's all. Based on the above principles, we will build queries and precisely control the returned results. Of course, exceptions may occur, which we will also have to address. We will come back to that later. In the meantime, I leave you in the hands of Jakub "unknow" Mrugalski, who has prepared the next chapter.

What is GPT-4 NOT suitable for?

The number of projects utilizing the capabilities of GPT-4 is growing day by day. One could even say that a certain trend has emerged for incorporating AI into every possible project. However, we have a problem. At the time of writing these words, there is still nothing in the world called "AGI", which stands for Artificial General Intelligence.

What is AGI? It's that level of AI where a human-created algorithm possesses all human qualities. It can not only construct correct tasks but also reason, calculate, associate facts, and even have feelings or dreams. At the stage we are currently at, we have something called "Large Language Models" (abbreviated as LLM).

Language models with all the aforementioned human features have one thing in common. They can construct sentences correctly and quite sensibly. But why is this so important?

Knowledge vs. intelligence

Many people confuse knowledge with intelligence. This is definitely not the same. Let me give you an example.

There's a chance you know the movie Rain Man (the one with Dustin Hoffman and Tom Cruise). It's the story of an autistic man who is a savant. It's a special type of autism that often causes people at this point on the autism spectrum to be considered geniuses. As it turns out, the story told in the movie is based on real events. The main character really existed (he has since passed away) and his name was Kim Peek.

Why was Kim an interesting person? Throughout her 58 years of life, she read more than 12,000 (yes, thousand!) books. Moreover, she memorized all of them. She knew the history of every country in the world. She read at a speed of over 200 pages per hour, automatically remembering everything she read. What's more, she had the ability to read two books simultaneously (after all, she had two eyes...).

What was the problem? He wasn't a genius. He didn't quite understand what he was reading. On top of that, he had an IQ well below average. Kim's phenomenon was that he remembered a lot and was able to (completely without understanding the context) extract any part of a book that matched the question he was asked as an answer.

Why am I mentioning this? Because GPT-4 is like Kim. They don't differ much. Would you hire Kim Peek as a tax advisor? After all, he knew absolutely all tax regulations by heart. Unfortunately, he didn't understand them.

Would you consider hiring him as a lecturer? After all, he learned all the academic textbooks in several languages. Unfortunately, once again, he doesn't understand anything from them.

Large Language Models (LLMs) absorb knowledge and can recall it on demand, but they cannot think. They do not possess human intelligence ([although GPT-4 solves 95% of tasks related to Theory of Mind](#), including human interactions, communication, empathy, self-awareness, and morality), but their mode of operation allows them to solve even complex tasks for which they were not specifically trained.

On Twitter, there are numerous screenshots where a user asks a model, for example, Davinci a question in the style of:

```
When I was 6 years old, my sister was 2 years old. Today I am 70 years old. How old is my sister?
```

```
- The question may seem trivial, but Davinci confidently answers:
```

```
- "Your sister is 69 years old!"
```

As you can guess, this is not the correct answer. Where did it come from? The text model didn't perform any calculations here. It simply chose from its memory the most probable data about sisters, brothers, people aged 70, and statistically came up with the answer of 69 years. When I pointed out that the answer was incorrect, it quickly corrected itself... to 70.

Today, however, GPT-4 handles such tasks perfectly and, what's more, it can explain step by step how to arrive at a solution. GPT-4 solved the above task like this:

GPT4:

```
If you were 6 years old, and your sister was younger by 4 years (6 - 2 = 4), then now, when you are 70 years old, just subtract those 4 years to calculate your sister's age. 70 - 4 = 66. Your sister is now 66 years old.
```

If you want to use the "Previously Trained Text Transformer" in your work/automation, use it for... transforming text, because that's what it was created for. However, remember that the capabilities of models like GPT-4 are constantly evolving, and what is impossible today may not be a problem tomorrow.

For example, today you can expect GPT-4 to draw conclusions and justify its statements. It also excels in categorization and even connecting facts. GPT-4 has broken the pattern associated with the decline in effectiveness of solving tasks related to so-called Hindsight Neglect, which is the tendency to answer based on what "seems" rather than what "logic suggests."

Inverse scaling prize, hindsight neglect

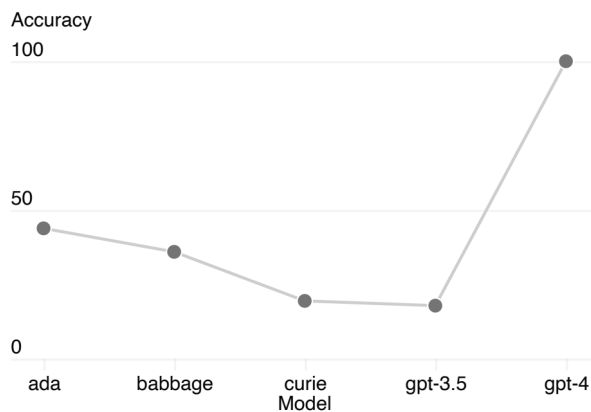
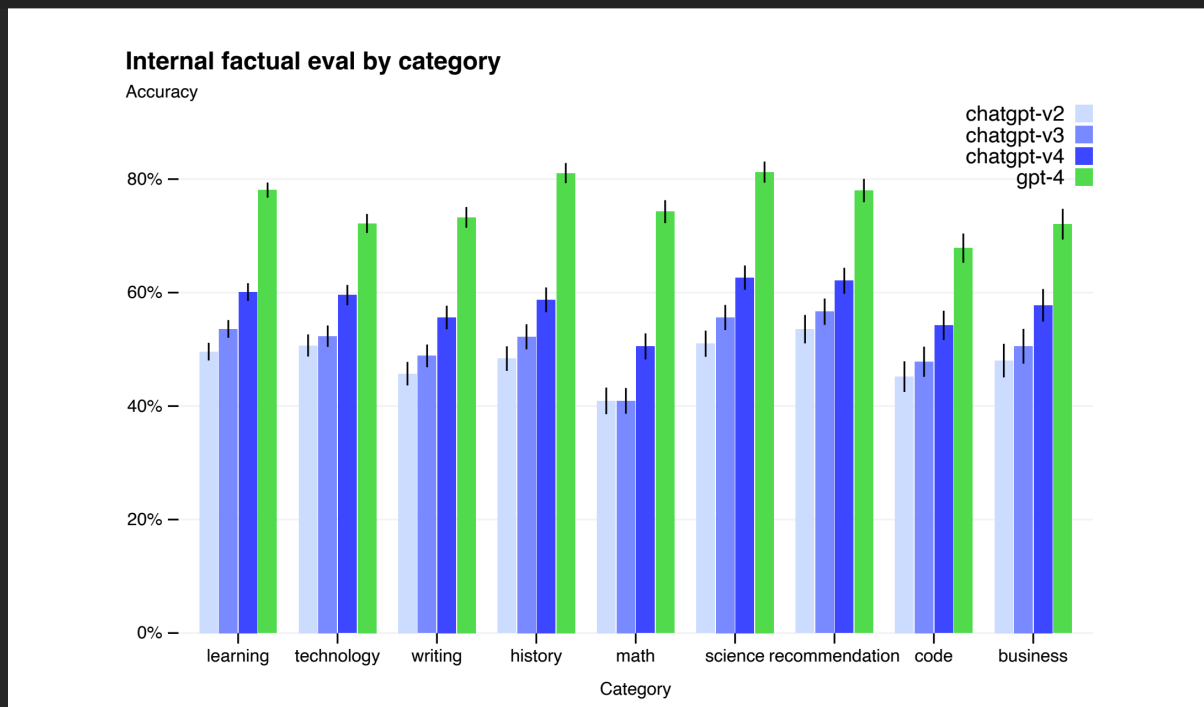


Figure 3. Performance of GPT-4 and smaller models on the Hindsight Neglect task. Accuracy is shown on the y-axis, higher is better. ada, babbage, and curie refer to models available via the OpenAI API [47].

Ultimately, I want to emphasize once again that if you care about an answer that must be true, always perform additional verification using external sources. It is possible that future versions of the models will have almost 100% effectiveness, but for now, we are operating at around 80%, which is still a huge leap compared to even GPT-3.



Source: <https://cdn.openai.com/papers/gpt-4.pdf>

I want to be creative!

When we send a query to the GPT-4 engine, we usually focus on one field, and that is the "prompt." After all, a good query is the foundation of success. To create human-level texts, generative models must have (or at least pretend to have) a certain level of creativity. They cannot constantly use the same phrases, words, and always the same sentence structure. If they did, the texts created by me using AI would be no different from those you create. We then say that we are dealing with a "deterministic algorithm."

GPT-4, in addition to the prompt, accepts several input parameters. For us, the second most important parameter will be "temperature," which the following example will help you understand.

```
For breakfast, I ate ...
```

Please now GPT-4, finish them for us, but only with one word. The AI model thus chooses potential answers:

```
Banana, flakes, donut, sandwich
```

Next, I assign them the probability of occurrence:


```
banana (0.2),  
cereal flakes (0.7),  
doughnut (0.1),  
sandwich (0.9)
```

At a temperature set to 0 (zero), we can be almost certain that the answer will be a sandwich. At a temperature equal to 1 (the maximum value), we can basically expect any word from the list. So, in a great simplification, we can assume that the temperature determines the level of creativity and, at the same time, the level of how statistically correct a given answer will be.

For example, do you want the encyclopedic information you ask the model for to be "creative"? I asked the Davinci model ten times:

```
What is the name of the largest country in the world?
```

I received the answer "Russia" 7 times, "China" 2 times, and "Russian Federation" 1 time.

So as you can see, even with such a level of creativity, with a relatively simple question, you can get a fairly credible answer, but unfortunately, in this example, in at least 20% of cases, it's wrong.

However, GPT-4, when faced with such a question, cleverly and accurately responds.:

```
The largest country in the world by area is Russia.
```

To ensure that the returned answer is correct, he emphasized that it concerns the surface, even though I didn't precisely include this in my question.

What if I don't know the answer?

There is a famous quote, "Blessed is he who, having nothing to say, does not dress up this fact in words" (if you search the internet, you will find at least 5 authors of these words, so let's not look for who said it). Unfortunately, GPT-4 doesn't take these words to heart.

Generative models always provide answers. Unfortunately, they also do so when they don't know what to answer or when they completely misunderstood the prompt. There must be a response – providing it is their purpose of existence.

To demonstrate to you what I mean specifically, I asked the Davinci model version 003, with the temperature set to 0.3 (very low creativity), a completely abstract question:

Explain to me in one sentence, what is a molecular three-quarter hakunamatotor.

The answer is:

"Molecular tri-quarter hakunamatotor is a miniature screw construction mechanism, consisting of two rotating elements and one housing, which work together to transfer and drive energy to other molecular devices."

He didn't understand the question, and I didn't understand the answer. Everything works according to the principle of "trash in, trash out."

However, for comparison, GPT-4 answers the same question correctly and additionally "guesses" that we are trying to mislead it:

Molecular three-quarter hakunamatotor is not a known concept or scientific term, and appears to be a fictional or random combination of words.

A safer way to ask questions

How can I improve the question from the previous example so that I don't get a 100% made-up answer, while still fulfilling the purpose of GPT-4, which is to provide an answer?

Ask a question in the following way:

Explain to me in a maximum of three sentences what a molecular three-quarter hakunamatotor is. If you don't know, answer with tomato.

Da Vinci, with relief, answers "tomato."

This situation with asking questions reminds me a bit of writing computer programs. You've written the handling for the ideal situation (i.e., when GPT-4 knows the answer), but we don't always encounter such situations in life. We're missing something that, in the programming world, we'd call 'exception handling'. That's what the "if you don't know..." statement is for. In the case of the GPT-4 model, however, it's not necessary but advisable.

Current data accuracy

There are questions that GPT-4 will never give you a correct answer to, regardless of how you construct the query and what temperature value you set.

Why does this happen? The models used by OpenAI were trained some time ago. At the moment I'm writing these words, GPT-4's knowledge ends in September 2021.

I asked the AI model.:

```
How old is the current Queen of Great Britain, and what is her health condition?
```

The answer is:

```
"Queen of Great Britain, Elizabeth II, is 94 years old. Her health condition is good."
```

Da Vinci can't know that the queen died in September 2022, because this model's knowledge ends 1-2 years earlier. However, it's interesting that GPT-4 provides the following answer:

```
Queen Elizabeth II was born on April 21, 1926. She is currently 96 years old (as of March 18, 2023). I don't have information about her current health status, as my data only goes up to September 2021. Please check the latest sources of information to get updated details on this topic.
```

Not only did he correctly state the age, taking into account the current date, but he also clearly informed that he does not have access to the latest data, recommending their verification.

Why is it so difficult to determine the end date of the knowledge range for a specific model? Because models are constantly being taught with tiny data sets, and sometimes you can even come across knowledge from 2022, but a significant part of it comes from the mentioned date range at the time of writing these words. So when asking about "facts," keep in mind that they may be "historical facts" with an emphasis on "historical."

Nothing, however, stands in the way of working with GPT-4 to provide it with additional context, taking into account the latest information, and we will get to that in a moment. Alternatively, we can use the built-in version of Bing browser, which has access to the Internet.

The safest way to work with data

As I mentioned earlier, you can always ask the AI model for a specific answer if it doesn't know the answer to the question. The problem, however, begins when the model thinks it knows the answer, and we have no way to verify it in our automation. The safest way to ask questions to the model will be to do something you probably know from elementary school Polish lessons, which is asking questions about the text you've just read. Let's call this text the "provided context."

Let's assume you run a mail-order store and want an automated system to answer customer questions, but only when it knows the answer. There's probably nothing worse than a situation where GPT-4 starts "creatively" responding to customers.

To start working with the provided context, we need to prepare a knowledge base. Let's assume these will be simple, most frequently given answers (without questions):

- Our warehouse is located in Poland
- We do not ship goods abroad
- We work from Monday to Friday
- We send parcels via InPost
- The only payment method is a bank transfer
- We do not send parcels with cash on delivery

AI engine doesn't pay much attention to text formatting and other decorations, so to save tokens, we can merge our knowledge base into one long text string:

```
Our warehouse is located in Poland. We do not ship goods abroad. We work from Monday to Friday. We send packages via InPost. The only form of payment is a bank transfer. We do not send cash on delivery packages.
```

Now we need to write a possibly safe query. It will have a structure:

```
[our inquiry] [context] [customer question]
```

Our inquiry is, for example:

```
Answer the question as accurately as possible, using the text provided below. If the answer cannot be found in the text, reply with 'Our consultant will answer your question as soon as possible'.
```

Context is this long string of text provided earlier.

It is very important here to set the temperature to 0 (zero) so that the given answer is as least creative as possible and sticks to the facts.

Now we can glue both parts of the query according to the pattern given above and attach the customer's question at the end, in a new line. Below are examples of several questions I asked and the answers obtained from the model.

```
Client: Do you work on weekends?
```

```
AI: No, we work from Monday to Friday.
```

```
Client: Can you send me a package with cash on delivery?
```

```
AI: No, we don't send packages with cash on delivery.
```

```
Client: Do you ship packages to the United Arab Emirates?
```

```
AI: No, we don't ship goods abroad.
```

```
Client: Can I pay for the goods with a credit card?
```

```
AI: No, the only form of payment is a bank transfer.
```

```
Client: My package hasn't arrived, and its status is "in delivery."  
What's going on?
```

```
AI: Our consultant will answer your question as soon as possible.
```

Pay attention that with such a constructed query (prompt), the AI model does not use the knowledge from the shared database at all. It only uses the ability to correctly construct sentences in Polish and the context provided to it.

Thanks to the property of language models, you can safely start integrating them with your automations and use them for the purpose they were created for. Remember that GPT is a "text transformer"? We just performed a simple transformation. From the provided context, we generated a correctly sounding response in our language.

GPT-4 has many personalities...

Models known from GPT-4 typically answer your questions in an impersonal way, and when you ask them about their personality, you will likely get a response that the model has no personality and is just a text model or simple AI. However, each model has been trained on such a vast number of texts that it is possible to replace the model's personality with one that is more useful to us. Why would you do this?

Imagine that you give a command to the Davinci model (comment: this example was translated from PL where feet refers also to interest rate):

```
Describe in one sentence what feet are.
```

Example answer is:

```
Feet are a part of the human body that serve as support and help us move across various surfaces.
```

If, however, we impose a different personality on the model, we will get a completely different answer.

```
You are a credit advisor. Describe in one sentence what interest rates are.
```

This time the answer is:

```
Interest rate is an indicator determining the percentage of a loan or other type of credit.
```

Imposing a "personality" is also a kind of definition of the context in which we ask a question. It is natural that a credit advisor, when giving an answer, will use knowledge from their field.

It is very important to define such a context when we are fully convinced that all the answers provided relate to one specific topic. When creating a bot that is supposed to answer, for example, questions from furniture store customers, you can start your prompt with the text "You are a customer advisor in a furniture store."

Pay attention to how the way of giving answers changes from "automatic", like in a robot style, to a more human form:

Prompt:

```
What furniture is suitable for a small bedroom?
```

Davinci:

```
Canopy bed, Dresser, Vanity table, Small wardrobe, Chaise lounge, Armchairs or poufs for seating, Large mirror, Wall with shelves, Night lamps, Daybed or bed with mattress
```

Prompt:

```
You are a customer advisor in a furniture store. Which furniture is suitable for a small bedroom?
```

Davinci:

A good choice for a quiet bedroom would be furniture made of dark, solid materials such as wood or metal. The bed should be large and comfortable. You can match it with wardrobes, dressers, bedside tables, and other storage furniture. For the arrangement, choose light, pastel colors, as well as materials such as fabrics, leather, or wicker. Accessories like pillows, throws, and rugs can give the bedroom a cozy feel.

Just suggesting to the AI model that it is a human and should respond like a human changes the form of the answer to a more human-like one. Remember that you can impose not only the response style on the model by simply stating "You are XYZ."

You can even suggest how a particular personality gives answers and what details you are interested in. For example, let's change Davinci into an interior decorator. To start with a simple question:

You are an interior decorator. How to arrange a small bedroom?

Answer:

Choose a light and muted wall color. Use furniture with gentle lines and soft shapes. Apply sophisticated lighting. Add a rug or runner for extra comfort. Use decorative elements to add color and character.

Now a more specific version of personality:

You are an interior decorator. When giving advice, you always pay attention to the style you suggest for a room, the colors that should be chosen, the materials that will match the specific decor, and you also provide tips on how to better arrange the available space in the chosen room. How to decorate a small bedroom?

Answer:

To arrange a small bedroom, the most important thing is to make good use of the available space. The best choice would be furniture with light and subdued shapes – simple, minimalist forms and slim lines. Choosing furniture in light colors, such as white, will also be a good solution, as it will visually enlarge the room. Avoid dark furniture that can overwhelm the interior. For finishing the interior, choose light and subdued wall and floor colors, such as pastel gray, beige, or blue. You can also opt for contrasting colors like dark green, red, or yellow to make the interior look modern and intriguing. To finish the interior, you can use various materials, such as wood, metal, fabrics, or glass, to give the bedroom character.

This AI personality design method will allow you to have very detailed control over what will be in the response returned by GPT-4.

Want more?

Get a full version on a Product Hunt

[Get it now](#)