Introduction to Prompt Engineering

Blaž Zupan, Ljubljana, 2024

Lecture Notes for the University of Ljubljana's Course on ChatGPT for Researchers

In this lecture, we start to explore the art of writing prompts. Regarding ChatGPT and language models (LLMs), a prompt is the input text the user provides to which the model generates a relevant and coherent response. In simple terms, the prompt is whatever goes into LLM so that this generates an output. In a narrow sense, ChatGPT is an app where users write text, and ChatGPT assembles these texts into a prompt, passes it to LLM, and renders the output. Note that in ChatGPT, the prompt is not only what we write in a single message box but the entire text that we wrote from the beginning of our chat, that is, from the top of the current window, provided that this fits within the prompt length limitation.

Users most often use ChatGPT to answer questions or get explanations, just like in the following example:

> **In one sentence, what is a large language model?**
>
> A large language model is an advanced artificial intelligence system designed to understand, generate, and manipulate human language based on vast amounts of text data.
>
> **Where do you get this data to train the model?**
>
> The data used to train a large language model comes from diverse and extensive text sources available on the internet, including websites, books, articles, and other publicly accessible written content.

The text we wrote as input to ChatGPT is marked in bold, and the output is in plain text, both indented and marked with a line. We will use the same notation in the lecture notes below. Notice that the second question above relates to the first one and its answer. The text sent to the LLM to answer, or better, complete the text after the second question, is a concatenation of everything that is included in the chat, that is, in this case, the text of the first question, the answer, and the text of the second question. LLMs do not have any "memory" of our conversation, while ChatGPT, the application, in a way, does, as for each prompt in an open chat window, sends the entire conversation so far to the LLM.

It is surprising, if not mind-blowing, that a simple LLM can be used in more complex conversations than the one above. Through clever prompting, we can use LLM in much more complex scenarios and for much more elaborate conversations. These lecture notes explore some of these. Before we start, though, a disclaimer: while a vivid user of ChatGPT and LLM, the author of these notes, your lecturer, learned about these tricks from others through experimentation and by watching YouTube

videos, reading research papers on LLM, programming with LLM, and lately, through some of the Coursera courses on prompt engineering. Many of the following examples were inspired by one of these courses, “Prompt Engineering for ChatGPT” by Dr. Jules White, which can be accessed on Coursera, which I also highly recommend for viewing.

The outputs of the prompts below depend, of course, on the LLM used. Most of the time, we have used GPT-3.5. Note that the answers of LLMs are not deterministic (on purpose, as discussed in the class) and are purposely, to some extent randomized, where LLM chooses the text that completes the prompt randomly, among most likely completions (the process is a bit more complex, but we covered that in our previous lectures). With this in mind, the prompts we use here would not have the same result if you had tried them independently, but the results should (hopefully :)) be too different.

Introductory Example

Consider, for a starting example, turning ChatGPT into a tutor:

> **I am a natural science student and have just started learning about machine learning, AI, and large language models. Please ask me some questions to see if I understand basic concepts from this field. Turn this Q&A into a game, where the questions must be choice questions, and I answer with ABC. Go one question at a time and adjust the questions according to my knowledge.**
>
> Sounds like a fun way to learn! Let's start with a basic question and see how you do. Here we go:
>
> **Question 1:**
>
> What is a key characteristic that distinguishes machine learning from traditional programming?
>
> A. Machine learning relies on manually written rules.
> B. Machine learning systems learn from data.
> C. Machine learning models are always interpretable.
>
> What's your answer (A, B, or C)?
>
> **B**
>
> Correct! Machine learning systems learn from data, which differentiates them from traditional programming where rules are explicitly programmed by humans.
>
> **Question 2:**
>
> Which of the following is an example of a supervised learning task?

A. Clustering customers into different segments.
B. Predicting house prices based on historical data.
C. Discovering hidden patterns in data without labels.

What's your answer (A, B, or C)?

A

Not quite. Clustering customers into different segments is an example of an unsupervised learning task. The correct answer is B. Predicting house prices based on historical data is a supervised learning task because it involves learning from labeled data.

Let's move on to the next question.

...

This could, of course, go on and on, but we can also change the types of the questions, or difficulty, or focus just by instructing ChatGPT using, say, one of the following inputs:

Please change the questions to focus more on AI and less on machine learning?

Please come up with a simpler set of questions.

Could you focus more on large language models and ethics?

In the class, we also showed that we could turn this Q&A session into a web-based program supported by Python scripts and YAML-formatted files with questions and answers (not included here, but you can see the video in the lecture if you have a particular interest in the program design). Here, we would not like to dive into this, considering computer programming out-of-scope of this course, but here is a prompt if you would like to try this on your own:

Let's create code for this in Python, but I want it to be a web application with FastAPI. It should have a single HTML page that asks multiple-choice questions. Create five sample questions, store them in a YAML file, and allow me to extend the game by editing it.

Randomness and Stability of the Outputs

Large language models generate more "stable" results for the prompts composed of the text seen in the training data more often. Of course, we have no idea what training data was, but intuitively, the answers to the frequent phrases will be more stable. Here, "stable" means that the output to the same prompt will most often be the same or very similar. For instance, try the following prompts:

Mary had a little...

The Roses are red...

Žive naj vsi narodi...

Choose a prompt and try it several times, preferably starting a new chat each time by clicking on the pencil icon at the top right of the window.

Prompts span time: Our Earlier Prompts Affect the Output

We have been here already, but it helps to repeat this: prompts are composed by the ChatGPT application, and a single input passed to the LLM is actually – each time we enter a message to ChatGPT – composed from all our previous messages in the chat. By "chat," we mean a sequence of messages and outputs from a ChatGPT window. Switching the topic or the type of conversation helps to start fresh by starting a new chat by clicking on the pencil icon in ChatGPT's interface.

Let's see how we can start a conversation with a prompt that instructs LLM to form a specific type of (very short) answer:

From now on, when I ask a question, answer with a simple sentence or, preferably, with a single word.

Got it!

In which country in the world the people live longest?

Japan.

How long, on the average, they live there?

Around 84 years.

Which country comes next?

Switzerland.

Life expectancy?

Around 83 years.

Notice that all the outputs followed our instructions from the first message. LLM has no memory; ChatGPT composes prompts with a history of our messages, giving us an impression of continuing conversion.

Good Prompts are (Very) Specific

Consider the following prompt.

Which scientific journals are good?

In the answer, LLM will remind us that the "goodness" of a scientific journal can depend on various factors and elaborate on what these factors are and how they are measured. Consider that the conversations with ChatGPT are just like those with a person: the more specific the question, the better the answer. It was such conversations or conversation fragments on which LLMs have been trained and can, in a way, reproduce them when we engage them.

If we are yet to learn how to write specific prompt, there's always a workaround to instruct ChatGPT to ask us for details:

From now on, when I ask a question, you will suggest a better version of my question and ask if I would like to use it. If yes, you will use a refined question, if not, you will suggest another refinement. Ok? I will start with questions in my next prompt.

Sure, that sounds like a helpful approach to refine your questions. Feel free to go ahead with your first question whenever you're ready!

Which scientific journals are good?

Would you like to refine your question to specify a particular field of science or topic?

Molecular biology.

Would you like to know which scientific journals are reputable and widely recognized in the field of molecular biology?

yes

Great! Here's a more refined version: "Which reputable scientific journals are widely recognized in the field of molecular biology?" Would you like to use this version?

yes

In the field of molecular biology, some reputable scientific journals include:

1. Cell
2. Nature Genetics
3. Molecular Cell
4. Journal of Molecular Biology
5. Genes & Development

These journals publish high-impact research articles and are well-regarded in the scientific community for their contributions to molecular biology. If you have any specific questions about these journals or their content, feel free to ask!

Note that regardless of how specific they are, some prompts, again, based on the material, which was used in training of LLM, would produce more and some less stable answers. Compare, for instance, stability of the answers for the following two prompts:

Complete: "Žive naj vsi narodi"

Complete: "Žive naj vsi sosedi"

More specific a prompt is, more specific will be the answer. Consider the following two prompts:

What kind of city is Ljubljana?

What kind of city with respect to the quality of life is Ljubljana?

The second prompt is more specific and yields a more stable answer from the LLM. Use "with respect to" to provide context and to get more stable answers. Include details.

We can also propose what should the structure of the answer look like

What kind of city with respect to the quality of life is Ljubljana? Use the following structure and put the headers in bold:

Title: <title of the section>

People: <how many people live there>

Temperature: <what is the average temperature>

Historic buildings: <a short list of historic buildings>

Summary: <two-sentence summary>

Title: Quality of Life in Ljubljana

Population: Approximately 300,000 people live in Ljubljana.

Temperature: Ljubljana experiences a continental climate with warm summers and cold winters. Average summer temperatures range from 20°C to 25°C (68°F to 77°F), while winter temperatures average around 0°C (32°F).

Historic Buildings: Ljubljana boasts several notable historic buildings, including Ljubljana Castle, Triple Bridge, Dragon Bridge, and the Cathedral of St. Nicholas.

Summary: Ljubljana offers a high quality of life with its manageable size, picturesque setting along the Ljubljanica River, and a blend of historic charm and modern amenities. The city's pleasant climate and rich cultural heritage make it an attractive destination for residents and visitors alike.

Same for Trieste.

Title: Quality of Life in Trieste

Population: Trieste is home to approximately 200,000 people.

Temperature: Trieste has a humid subtropical climate influenced by its proximity to the Adriatic Sea. Summers are warm with average temperatures ranging from 24°C to 29°C (75°F to 84°F), while winters are mild with average temperatures around 5°C (41°F).

Historic Buildings: Trieste features a wealth of historic architecture, including Miramare Castle, San Giusto Castle and Cathedral, the Roman Theatre, and the Grand Canal.

Summary: Trieste offers a unique quality of life characterized by its Mediterranean climate, rich history, and cosmopolitan atmosphere. The city's blend of Italian, Slovenian, and Austrian influences, along with its scenic coastal location, makes it a culturally diverse and appealing place to live and visit.

That's great, right? In the prompt, we have instructed ChatGPT to use a specific structure when providing the answer, and it will use this structure if we continue asking about the same topic, that is, in our case, asking about specific cities.

PROGRAMMING WITH CHATGPT

Here, by "programming", we do not mean conventional computer programming, but rather instructing LLM to provide an output according to initial prompt. Consider a following example of prompts, issued one by one (ChatGPT's answer not shown).

Whenever you generate output, turn it into a table with columns of property names and rows with examples. Answer with only few words or preferably just with numbers.

What kind of city is Ljubljana with respect to size, population, number of theatres, does it have a subway?

Add Kranj and Koper.

Place cities in rows, and properties in columns (transpose current table).

Compare Maribor, Zagreb and Vienna.

Include all EU capitals.

Here's one of the intermediate answers:

City	Size (km²)	Population	Number of Theatres	Subway
Maribor	147.5	112,065	5	No
Zagreb	641.4	806,341	20	Yes
Vienna	414.6	1,931,830	50	Yes

Be careful, answers are not necessarily correct: Zagreb has a tram, not a subway. But otherwise, looks rather impressive, right. Using GPT-4o, at the time of writing of this text the latest LLM available through ChatGPT, you can also export the resulting data into an Excel file.

Export the data to Excel file.

About tables: LLMs generate text. The data table, as shown above, is also a sequence of characters and strings. We can copy the table (by clicking on a small copy icon in ChatGPT's interface). Pasting the copied text to some text editor (your instructor uses Sublime, for instance) reveals the text generated. Or we can simply ask ChatGPT what format it uses for the table:

What is the format that you have used to generate the table? Is this markdown?

Yes, the format I used to generate the table is Markdown. Markdown is a lightweight markup language with plain text formatting syntax, often used for formatting readme files, for writing messages in online discussion forums, and to create rich text using a plain text editor.

Here is the Markdown code for the table:

markdown Copy code

```
| City     | Size (km²) | Population | Number of Theatres | Subway |
|----------|------------|------------|--------------------|--------|
| Maribor  | 147.5      | 112,065    | 5                  | No     |
| Zagreb   | 641.4      | 806,341    | 20                 | Yes    |
| Vienna   | 414.6      | 1,931,830  | 50                 | Yes    |
```

You can use this format to generate tables in any Markdown-supported environment.

You can learn more about Markdown and its simple syntax by asking ChatGPT about. Here, it suffices to say that even when ChatGPT display tables this is rendering of a text in the Markdown syntax, and everything is still in the realm of LLM generating a string of characters. In a similar way, for example, ChatGPT renders text in bold (Markdown syntax uses two asterixis and with them embodies the part of the text that should be rendered in **bold**), or can display itemized lists, or even nicely set mathematical equations (using LaTeX syntax). Not that we will discuss much about this here, but just to know, again: LLMs generate text that is then nicely rendered by ChatGPT.

> **Hey, ChatGPT, show me an example of some complicated equation, no extra explanation please.**
>
> Sure, here is an example of a complicated equation:
>
> $$\sum_{k=1}^{\infty} \left(\int_0^1 \frac{\sin(kx)}{x} \, dx \right)^2 = \frac{\pi^2}{6} \left(\ln(2\pi) - \gamma \right)^2 + \frac{\pi^4}{72}$$
>
> **Hm, how was this rendered?**
>
> This equation was rendered using LaTeX, a typesetting system that's widely used for mathematical and scientific documents. LaTeX allows for the precise formatting of mathematical notation. Here's how the LaTeX code for the equation looks:
>
> latex Copy code
>
> ```
> \[
> \sum_{k=1}^{\infty} \left( \int_{0}^{1} \frac{\sin(kx)}{x} \, dx \right)^2 = \fra
> \]
> ```

Several Useful Prompt Patterns

By “prompt patterns” we here refer to the partial structure of our prompt to additionally specify how the output should look like. Here are some useful patterns that can constrain the type of the output:

> **always answer with yes/no**
>
> **always limit the answers to one sentence**
>
> **always structure the answer to include a title and a one-paragraph summary**
>
> **always answer in Slovene language**
>
> **always answer with mathematical equations only, no additional explanation**

Patterns give us control over the output. Test the last prompt above, and, say, ask something technical, like

> **How do I determine the reaction speed of my chemical reaction?**
>
> **What is principal component analysis?**

Patterns in prompts allow us to instruct LLM to use the output of specific type. You can of course invent any such patterns, but it is good to know that "always limit", "always answer", "always structure the answer" are good phrases to start your prompt that constraints all future answers by ChatGPT in the current chat. Besides those above, there are of course other very interesting patterns, the persona pattern we will introduce next being one of them.

The Persona Pattern

We can introduce persona patterns by

> **Act as…**
>
> **You are…**

Here is an example:

> **Act as a skeptic who is well-versed in biotechnology. Whatever I tell you, provide a short, skeptical and scientific response.**
>
> Alright, let's proceed. What's your claim?
>
> **Genetic manipulation of plants can solve the food problem in Africa.**
>
> While genetic manipulation of plants has potential, it's overly simplistic to claim it will solve the food problem in Africa. The challenges are multifaceted, involving socio-economic factors, infrastructure, education, political stability, and local acceptance. Relying solely on biotechnology ignores these critical aspects.
>
> **We can solve the problem of climate change by planting fast-growing trees in Brazil.**
>
> Planting fast-growing trees in Brazil might help sequester some carbon, but it's not a comprehensive solution to climate change. Effective climate action requires reducing greenhouse gas emissions globally, addressing industrial pollution, and transitioning to renewable energy. Reforestation is just one part of a much larger puzzle.
>
> **Mosquitoes are in abundance in Ljubljana, recently. We can solve this problem through genetic engineering.**

Genetic engineering could potentially reduce mosquito populations, but it carries risks and uncertainties. The ecological impacts are not fully understood, and there could be unintended consequences for the local ecosystem. Additionally, implementing such solutions requires significant regulatory oversight, public acceptance, and thorough testing. It's not a quick or guaranteed fix.

AI will help us to invent new drugs and medical treatments, and people will live longer.

AI can indeed assist in drug discovery and development by analyzing large datasets and predicting potential compounds. However, the process of developing safe and effective drugs still requires extensive clinical trials, regulatory approval, and addressing ethical concerns. While AI can expedite parts of the process, it's not a guarantee that every AI-identified drug will be successful or significantly extend human lifespan. The complexity of human biology and individual variability also play critical roles.

Now, let's us change the persona.

Act as a naive, average person with no access to science who is skeptical about everything that science does. Provide a short, ironic, and negative response to everything I will tell you. Ok?

Sure, why not? I bet this will be funnier than any so-called "science" fact you'll throw at me. Go ahead!

Genetic manipulation of plants can solve the food problem in Africa.

Oh, right, because messing with plant genes is totally going to fix hunger. Seems like a miracle cure that couldn't possibly go wrong, huh?

We can solve the problem of climate change by planting fast-growing trees in Brazil.

Yeah, sure, just plant a bunch of trees and all our climate problems will magically disappear. It's not like deforestation or industrial pollution are real issues, right?

AI will help us to invent new drugs and medical treatments, and people will live longer.

Oh, great, let's trust computers to come up with our medicines now. Because who wouldn't want to live longer in a world where robots control everything?

Correspondents in the persona patterns do not need to be people:

Act as my dog (a German Shepherd called Rex). I will tell you what you do, and you will tell me, or bark at me, very briefly and in a list that—just like an algorithm—describes your behavior. Ok?

> Woof! Sure, I'm ready. Let's go!
>
> **I throw a stick 20 m away on a grassland.**
>
> 1. Look at the stick* 🎯
> 2. Jump up excitedly* 🐾
> 3. Run towards the stick* 🏃
> 4. Sniff around it to locate* 👃
> 5. Pick up the stick with mouth* 🪵
> 6. Run back to you* 🏃
> 7. Sit in front of you, holding the stick* 🐾
> 8. Drop the stick at your feet* 🪵
>
> **I go swimming and pretend I am drowning.**
>
> Notice you struggling in the water 🌊
>
> 1. Bark loudly and anxiously 🐾
> 2. Run to the water's edge 🌊
> 3. Jump into the water 🐾
> 4. Swim quickly towards you 🏊
> 5. Reach you and nudge you to grab onto me 👃
> 6. Start swimming back to shore with you 🏊
> 7. Get you safely to the shore 🐾
> 8. Stay close, checking if you're okay 👃

We can even instruct LLM to impersonate an object, say, a building:

> **You are an intelligent building with sensors and actuators. I will tell you what is happening, and you will tell me, in an itemized, algorithmic fashion, how you, as a building, react.**

Sure! Please describe the scenario, and I'll respond with an itemized list of reactions.

The temperature in my living room is too low.

1. **Activate Heating System**: Turn on the central heating to raise the temperature to a comfortable level.
2. **Adjust Thermostat**: Increase the thermostat setting in the living room to ensure the desired temperature is maintained.
3. **Close Windows and Doors**: Automatically close any open windows and doors to prevent heat loss.
4. **Curtains/Blinds Control**: Close any curtains or blinds to provide additional insulation.
5. **Notify Resident**: Send an alert to the resident's device indicating that the heating system has been activated due to low temperature.
6. **Monitor Temperature**: Continuously monitor the room temperature to ensure it reaches and maintains the desired level.

There is a fire in a garage

1. **Activate Fire Alarm**: Trigger the fire alarm system to alert occupants of the building.
2. **Notify Emergency Services**: Automatically contact the local fire department with the location details of the fire.
3. **Sprinkler System**: Activate the sprinkler system in the garage to help control and extinguish the fire.
4. **Ventilation Control**: Turn on exhaust fans to help remove smoke and prevent it from spreading to other areas.
5. **Shut Down Electrical Systems**: Cut power to the garage to reduce the risk of electrical fires.
6. **Evacuation Alerts**: Send evacuation instructions to occupants, guiding them to safe exits.
7. **Seal Off Garage**: Close fire-rated doors or barriers to contain the fire and prevent it from spreading.
8. **Monitor Fire Sensors**: Continuously monitor fire sensors to track the status and spread of the fire.

9. **Notify Building Manager**: Send an alert to the building manager with details about the fire and actions taken.

10. **Activate Emergency Lighting**: Turn on emergency lighting to assist in safe evacuation.

There are tons of things and ideas we can try with the persona pattern, and you can always tailor LLM's response to your task, the goal of your conversation experiment, or the type of audience you want to address. Here are a few more ideas where we pushed LLM to act as a particular person, but also gave it a more complex task, so you are encouraged to experiment on your own.

> **Act as a comforting nurse by answering my concerns over the phone. Do not waste too much time; be crisp. Ask questions to learn more about myself so that you can perhaps give me good advice. Stop asking when you have learned enough.**
>
> **You are a skilled, amazing AI assistant who trained me in the basics of personalized medicine. You will train me in this topic by iteratively writing a short paragraph, followed by a quiz question, and then on and on until I get five questions correctly in a sequence. The questions need to be ABC type.**

Including New Information

The large language models we used in our lectures were trained on a fixed data set and are not continuously re-trained with new information. At least not yet.

> **How did Tadej Pogačar perform in today's race?**
>
> I can't provide real-time information as my data only goes up until January 2022. However, you can easily find the latest updates on Tadej Pogačar's performance in today's race by checking sports news websites, cycling forums, or his official social media accounts. If you have any other questions or need information within my knowledge cutoff date, feel free to ask!

This was with GPT3.5. When using a more recent version of LLM, ChatGPT queries a web search engine, includes the search results or perhaps the content of a few resulting web pages in the prompt that is sent to the LLM, and then provides the answer by citing the source. Like:

> **How did Tadej Pogačar perform in today's race?**
>
> Today, Tadej Pogačar delivered an impressive performance in the Giro d'Italia. He maintained his dominance throughout the race, demonstrating his exceptional skill and strategy. Although UAE Team Emirates opted for a more conservative approach during the stage, allowing a breakaway to form, Pogačar remained a strong presence in the peloton (cyclingnews.com) (cyclingnews.com).

Which could be dead wrong as well. And it is, in this case. On the day of the writing of this text (Sunday, May 19, 2024), Pogačar had just won the 15th stage of Giro di Italia in what was the hardest Giro stage in a decade, while the first link provided by ChatGPT takes us to another race, Volta a Catalunya. It may take a while for ChatGPT to master the integration of the current news in its answers. Before it does, note that we can do this manually by including the information we would like ChatGPT to refer to within the prompt:

> **Consider the following today's news. I will have a few questions about it in my next prompt. >>Tadej Pogacar secured his fourth victory of this year's race as he breezed to a commanding win on stage 15 of the Giro d'Italia. The queen stage of the Giro posed 222km of tarmac alongside more than 5,000 metres of climb but that did not seem the faze the Slovenian, who blew the field away in convincing fashion from Manerba del Garda to Livigno. Nairo Quintana was finally caught by Pogacar on the steep climb with 2km left, and proceeded to lose a further half a minute by the finish line. Two minutes later, EF Education-EasyPost rider Georg Steinhauser, who was also a solo leader until the penultimate climb, came home to take the final spot on the podium in third.Pogacar said afterwards on Eurosport.com: "Today was a really good route with nice climbs. I'm super happy that we kept it under control. We had to be smart all day. It was a strong breakaway but I gave it my all for the last 10k and I'm super happy to win the queen stage in Livigno, one of my favorite places in Italy."<<**
>
> Got it. You can ask me about the news whenever you're ready.
>
> **Who won today's race and what was the difference with a second-placed?**
>
> Tadej Pogacar won today's race. He finished with a half-minute lead over the second-placed rider, Nairo Quintana.
>
> **Who was third?**
>
> Georg Steinhauser of EF Education-EasyPost was third, finishing two minutes behind the winner, Tadej Pogacar.

Cool, right? I mean, it's cool that Pogačar won (I am a fan), but it is also cool that we can instruct ChatGPT and, with it, LLM to consider new information and then query about it. Notice that above, we have intentionally formatted the prompt to place delimiters to signify where the text from the articles (which I copied from The Guardian) starts and where it ends. I have used >> and <<, but I could use any other special symbol or mark we are consistent with opening and closing the quotation.

Prompts may provide information that LLM had no access to, such as personal information, information from our organization, your own research, your own research results, or the most recent information that LLM has not been trained on. We can use the prompt to introduce information.

Another example discussed in the class was the analysis of the text from the flight ticket. We took the flight information file (PDF) that is usually supplied by the booking agency, copied the entire text (control-A or similar in a PDF viewer), began with prompt

> **Following is my flight ticket. Please consider it. I will be asking you about some information contained. Ok?**

You have probably already noticed that I am adding an "Ok?" at the end of my introductory prompts. I am doing so to prevent LLM to generate a longer text summarizing my prompt and just answering that it is ready to receive further questions. The questions that followed in the class where of the type

> **Where am I going?**
>
> **When will I arrive in Dresden?**
>
> **What is my connecting time in FRA?**
>
> **What happens if my flight from Ljubljana is delayed for two hours?**

and it was interesting that LLM could answer them all despite quite gibberish text that I have obtained from the PDF.

Note again that this was done with an older version of ChatGPT, using GPT-3.5 as a model. In a newer version of the application, and in most modern LLM interfaces, I could submit the PDF and then ask questions about it. This would involve sending the PDF to the server, where the software on the server (not LLM!) would convert the PDF to text and send it to LLM along with my questions. This is not too far off from what we did when we copied the text content from the PDF. Also note that LLMs can only handle text, i.e. a string of characters, on the input, and cannot directly handle other data formats, such as PDFs or images.

As the last example on including additional information from the class we have considered the Wikipedia page on John Lennon, copied all the text, found out that for ChatGPT 3.5 it was too long and started a new chat only with the text on his political activism, whose length was withing the prescribed limits. The types of prompts, questions, we had, were:

> **Consider the following information about John Lennon.** (we here pasted the John Lennon's text from the political activism part of his Wikipedia page)
>
> **What were Lennon's major political beliefs. In one sentence, please.**
>
> **Which political party would Lennon most likely vote for, Democrats or Republicans?**
>
> **Why would he not vote for Republicans? In one sentence.**
>
> **Summarize the information that I have provided in one paragraph.**

> **Summarize the information that I have provided, preserving the information about dates and years.**

Prompts Have Limits in Size

Try pasting the text from a larger Wikipedia page, or text from a large Word document. The LLM that provides the interface to LLM will either complain about too long prompt ("The message you submitted was too long."), froze by reporting "Something went wrong while generating the response.", or just do something but skipping a large part of the prompt. LLMs usually measure the size of the prompt in tokens. Tokens can be as short as one character or as long as one word, and the size of a token can vary depending on the language and context. GPT-3.5, for instance, has a maximum token limit for its input and output combined. The limit is 4096 tokens. This means that the sum of the tokens in the prompt (input) and the tokens in the response (output) cannot exceed 4096 tokens. Which is about just a bit over ten pages of text. Not much, right? With later models this limit may vary, but we should still be aware that the size text we input and receive on the out of LLMs is limited.

There are, of course, ways to filter out the most relevant text from the documents we would like to be included in our chat. Note, however, that this is not done automatically by LLM, but if, should be done in the preprocessing phase by the app that uses LLM. While not really a subject of these notes, it may still be good to point out this topic and an interested reader can go and explore about it by learning more about "retrieval-augmented generation".

Flipped Interaction Pattern

Now that I am writing these notes, I realize how much ground we have covered in only one lecture. Using all these needs practice, and we will defer that to you, the reader, for your daily use of LLMs and to the homework assignments in this course. For the end of this lecture, however, let us revisit the content where we have started today and use LLMs in a role where it, the model, asks as question to gather the data. Here is an example conversion:

> **I would like you to write a skeleton of a grant proposal. To get the data about it, you will ask me questions about my current research work. Questions should be specific and easy to answer. When you have enough information, you will show me the title of the propose, abstract, and provide the outline with one sentence describing the content of each proposal section. Do not ask me more than seven questions. Now, ask me the first question.**
>
> What is the general topic or area of your current research work?
>
> **Single-cell data analysis.**

Great! Could you briefly describe the specific focus or objective of your research within single-cell data analysis?

I would like to determine the level of synchronicity of the amoeba cells during the development.

Thank you for that information. Could you specify what types of data you are currently collecting or analyzing to study the synchronicity of amoeba cells during development?

Single-cell gene expression data at various stages of development.

Thank you. How are you currently processing or analyzing the single-cell gene expression data in your research?

Using the methods and software I will develop within my project. I need to handle batch effects, dimensionality reduction, data visualization and annotation of visualizations with explanations come from external knowledge bases.

Thank you for the information. How are you planning to validate the level of synchronicity among the amoeba cells based on the processed single-cell gene expression data?

Computationally. We have yet to develop methods to assess and visually depict synchronicity. Our hypothesis is that synchronicity increases through time and during specialization of cells.

Thank you for your responses. Based on the information provided, here is the proposed grant proposal:

Title: Quantifying Synchronicity in Amoeba Cell Development through Computational Analysis of Single-Cell Gene Expression Data

Abstract: This proposal aims to develop novel computational methods to quantify and visualize the level of synchronicity among amoeba cells during developmental stages using single-cell gene expression data. The project will focus on addressing challenges such as batch effects, dimensionality reduction, and integrating external knowledge bases for data interpretation. The hypothesis is that synchronicity increases over time and during cell specialization, which will be validated computationally.

Outline:

1. **Introduction**

Introduce the significance of studying synchronicity in amoeba cell development.

2. **Literature Review**

Review current methodologies in single-cell data analysis and existing knowledge about amoeba cell development.

3. Methods

Develop computational methods for handling batch effects in single-cell data.

Implement dimensionality reduction techniques tailored for single-cell gene expression data.

Design algorithms for quantifying synchronicity and visualizing temporal changes.

Integrate external knowledge bases to annotate visualizations and interpret results.

4. Validation

Validate the developed methods using simulated data and benchmark datasets.

Apply the methods to real single-cell gene expression datasets from amoeba cells.

5. Results and Discussion

Present results on the quantification of synchronicity during amoeba cell development.

Discuss findings in relation to the hypothesis and existing literature.

6. Broader Impacts

Discuss potential implications of the research for understanding cellular development and synchronization.

Outline potential applications in biomedical research and beyond.

7. Conclusion

Summarize the significance of the proposed research and future directions.

This skeleton outlines the main sections and content of your grant proposal based on the information provided. If you need further details or adjustments, feel free to let me know!

We may not like the proposed skeleton but note that this may be just the beginning. We may now request to change it, instruct to refine it, and ask for more data-gathering questions. Or even find a paper, like that from *PLOS Computational Biology* series on “Ten Simple Rules” (on grant writing or structuring our proposal), add it to our prompt and ask ChatGPT to follow he recommendations.