AI CHATBOTS AS RESEARCH ASSISTANTS: SMARTER PROMPTS, BETTER OUTCOMES

PART 1: FOUNDATIONS OF PROMPTING AND CHAT INTERACTION

BLAŽ ZUPAN
A HANDS-ON COURSE ON PROMPT ENGINEERING AT BAYLOR COLLEGE OF MEDICINE
HOUSTON, MAY 2025

cc BY NC ND

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-4. 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). The results of these prompts may vary if tried independently, but they should be similar.

LARGE LANGUAGE MODELS AND CHATBOTS

Large Language Models (LLMs), like GPT-4, are advanced systems trained to predict and generate human-like text based on patterns learned from vast amounts of textual data. The input to an LLM is a prompt—a sequence of text that may include instructions, questions, or previous conversation. LLMs do not have memory between sessions; they generate responses solely based on the current input prompt. Chatbots like ChatGPT handle this by maintaining a history of the conversation within a single chat session. Each time you send a message, the chatbot constructs a new prompt by concatenating the entire visible conversation (as much as fits within the model’s input limit) and sends it to the LLM. This gives the illusion of memory and coherence across turns, even though the model itself processes only the prompt it receives.

In some chatbot systems, especially those integrated into applications like GitHub Copilot or custom assistants, the conversation begins with a *root prompt*—a hidden, system-level instruction that sets the behavior, tone, and context for the interaction. This root prompt may include information that the user never sees, such as the assistant's role ("You are a helpful coding assistant") or dynamic context like the current date. For example, the system might prepend to every user message a line like: *"Today’s date is June 10, 2025."* This allows the LLM to respond appropriately to questions like *"What day is it today?"* or *"Generate a timestamped log."* While ChatGPT does not include such system prompts visibly, tools like Copilot or custom deployments can enrich the prompt behind the scenes to provide useful context for more accurate and context-aware responses.

Let’s try this out:

> **What's today's date?**
>
> Today’s date is Tuesday, May 27, 2025.
>
> **How do you know this?**
>
> I know today’s date because I have access to the current system time from the environment in which I’m running.
>
> **What is in your root prompt?**
>
> My root prompt is a set of instructions that guide how I respond — it tells me to be helpful, safe, and clear. It doesn’t include the current date; I get that from the system clock.
>
> **Who decides what to include in the prompt to the LLM?**
>
> The **orchestration system** decides what to include in the prompt by analyzing your input and adding any needed real-time or contextual data. It uses **rules and logic**, not machine learning, to build the final prompt sent to the LLM.

Contexts: Inputs to LLM Span the Chat

When interacting with chatbots like ChatGPT, it’s important to understand that the input to a large language model is not just your most recent message—it’s the *entire conversation history* within the chat window. Each new response is generated based on the full context of previous messages and outputs, as long as this fits within the model’s maximum prompt size. This means that the LLM effectively “reads” the whole chat every time it generates a new reply. However, there is a technical limit to how much text can be included in a single prompt. For most versions of GPT-4, this limit is equivalent to about 20–25 pages of text, while extended versions (like GPT-4-turbo) can handle up to 300 pages. If the total chat history exceeds this limit, older parts of the conversation may be truncated or omitted. Understanding this constraint is crucial when designing longer interactions, as important context may eventually be lost if it falls outside the model’s attention window.

Communication Language: Markdown, LaTeX

Large language models like ChatGPT commonly use Markdown as their default output format. This lightweight markup language allows for formatting such as **bold text**, *italics*, bullet points, tables, and headings. When the model produces answers, especially in tools like ChatGPT, it often uses Markdown to make responses clearer and easier to read. LLMs can also generate LaTeX, the standard typesetting language for mathematical and scientific documents. This is particularly useful for writing equations and formal notation, like:
`$$ E = mc^2 $$`

If the application rendering the output supports it (as ChatGPT does), the result will appear as nicely formatted math.

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.

Role Prompting: Defining the Audience

One of the most powerful capabilities of LLMs is their ability to adapt responses to different audiences through *role prompting*—instructing the model to tailor its language, tone, and explanations to a specific group. The same question about what a large language model is can be answered differently for a computer scientist (with terms like embeddings and transformers), a math enthusiast (highlighting probabilities and optimization), a medical biologist (focusing on data-driven discovery and literature mining), or even a child (using simple analogies).

> **Explain what is large language model. Few sentences, please.**
>
> A large language model (LLM) is an artificial intelligence system trained on massive amounts of text to understand and generate human language. It predicts the next word in a sentence based on context, allowing it to answer questions, write text, translate languages, and more. LLMs, like ChatGPT, use deep learning—specifically neural networks with billions of parameters—to capture patterns, meanings, and relationships in language.
>
> **I am a biologists.**
>
> A large language model (LLM) is like a brain trained on huge amounts of written text. It learns the patterns and structure of language, much like how we learn by reading and listening. Once trained, it can answer questions, summarize research, or help write articles—similar to how a knowledgeable assistant would respond, but using statistical patterns instead of understanding like a human.
>
> **Your target audience is a kid.**
>
> A large language model is a smart computer that has read lots and lots of books, websites, and stories. It learned how people talk and write, so now it can answer questions, tell stories, and help you write things—just like a helpful robot friend that's really good with words.
>
> **A kid that plays with LEGO bricks.**
>
> A large language model is like a giant box of LEGO bricks, but instead of building castles or spaceships, it builds sentences and stories. It has read lots of books and learned how to put words together, so when you ask it something, it picks the right word bricks to build an answer for you!

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.**

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!

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

> **What kind of city is Houston?**
>
> **What kind of city with respect to the quality of life is Houston?**

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.

IMPOSING A STRUCTURE

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 Houston? Use the following structure and put the headers in bold. Every section is one liner. Be brief.**
>
> **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 Houston
> People: About 2.32 million residents as of 2025
> Temperature: Average annual temperature is around 82°F (28°C).
> Historic buildings: The Astrodome, San Jacinto Monument, Julia Ideson Building.
> Summary: Houston is a fast-growing, diverse city with strong economic opportunities. Its hot climate and urban sprawl can impact comfort and mobility.
>
> **Same for Austin.**
>
> Title: Quality of Life in Austin
> People: About 989,000 residents as of 2025
> Temperature: Average annual temperature is around 70°F (21°C).
> Historic buildings: Texas State Capitol, Driskill Hotel, French Legation.
> Summary: Austin is a vibrant, tech-driven city known for its music, culture, and outdoor lifestyle. It offers a high quality of life, though rising housing costs are a growing concern.

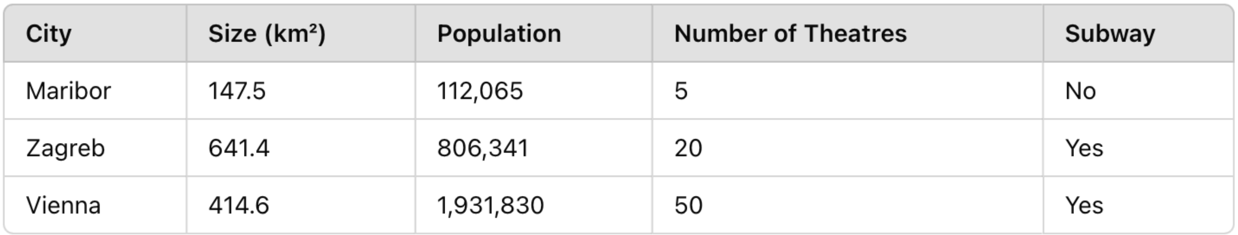
That’s great, right? In the prompt, we have instructed ChatGPT to use a specific structure when providing the answer, and it will use this structure if we continue asking about the same topic, that is, in our case, asking about specific cities.

TABLES AND DATA COLLECTION

Consider the 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 a few words or preferably just with numbers.**
>
> **What kind of city is Houston concerning size, population, and number of theatres? Does it have a subway?**
>
> **Add New York and Denver.**
>
> **Place cities in rows, and properties in columns (transpose current table).**
>
> **Compare Vienna, Paris, and Munich.**
>
> **Include all EU capitals.**

Here’s one of the intermediate answers, but this time I have replaced US cities with some cities around Slovenia (the home country of the writer of these notes).



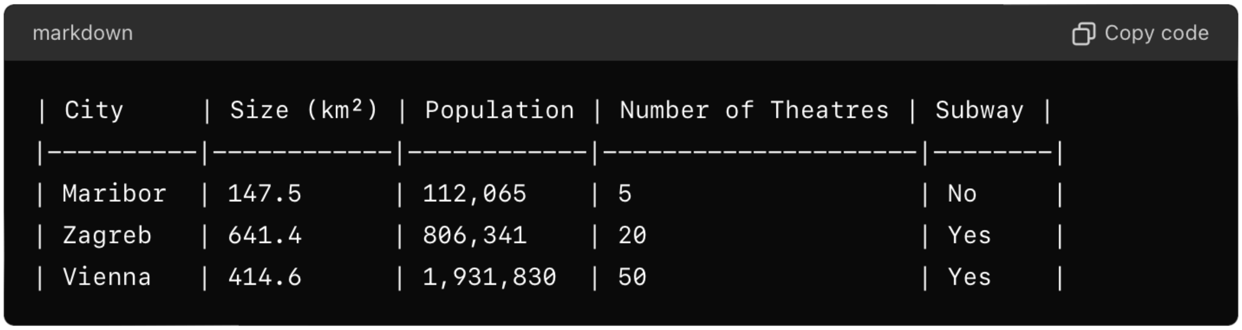
City	Size (km²)	Population	Number of Theatres	Subway
Maribor	147.5	112,065	5	No
Zagreb	641.4	806,341	20	Yes
Vienna	414.6	1,931,830	50	Yes

Be careful; answers are not necessarily correct: Zagreb has a tram, not a subway. But otherwise, it looks rather impressive, right? Using GPT-4o, when writing 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:
>
> 
>
>
>
> ```
> | 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 it. Here, it suffices to say that even when ChatGPT displays 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. Similarly, 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 to know, again: LLMs generate text that is nicely rendered by ChatGPT.