

Exploring Large Language Models

ChatGPT - What's the big deal?

Nils Vreman

December 13-15, 2023

Presentation Structure

- ▶ Every slide has been generated using ChatGPT
- ▶ Every image has been generated using DALL-E (one shot)
 - ▶ DALL-E is a modern text-to-image systems
 - ▶ Captions are summaries of the query used to generate the image
- ▶ If anything has been changed, it's marked in red

What is AI?

- ▶ Convolutional Neural Networks
 - ▶ Generative Adversarial Networks
 - ▶ Deep Neural Networks
 - ▶ Recurrent Neural Networks
 - ▶ Reinforcement Learning
 - ▶ Statistical Models
 - ▶

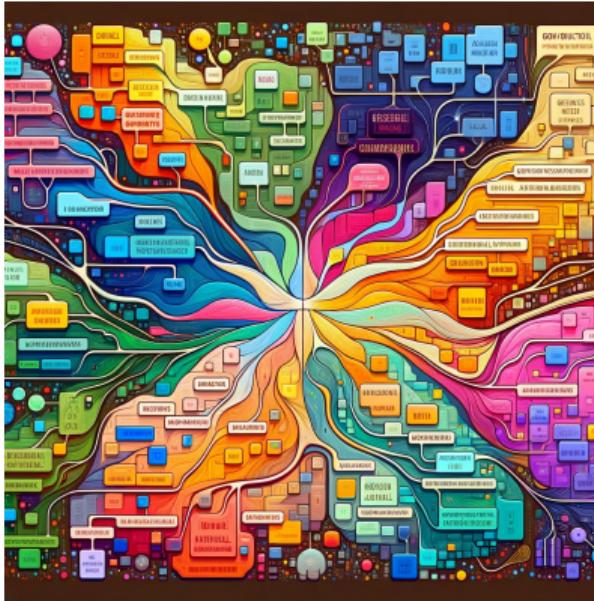


Figure: AI topology

What is Natural Language Processing?

Natural Language Processing (NLP) is a branch of AI that focuses on the interaction between computers and humans through natural language.

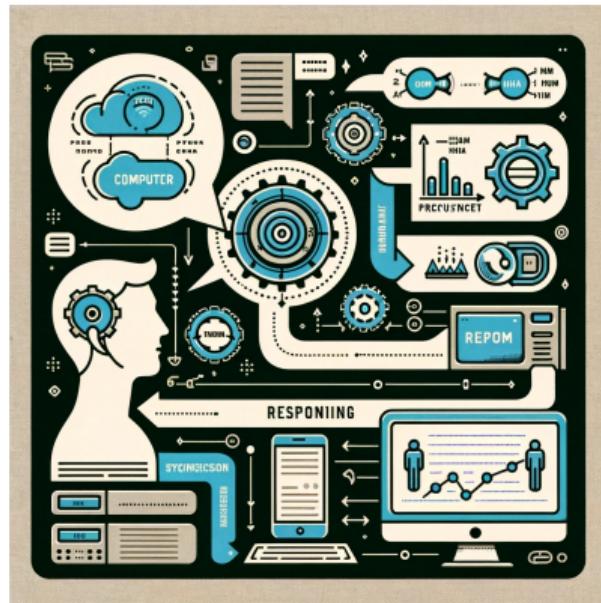


Figure: Infographic showing human ↔ computer interaction

Introduction to Language Models

Language models are AI models that understand, interpret, generate, and respond to human language.

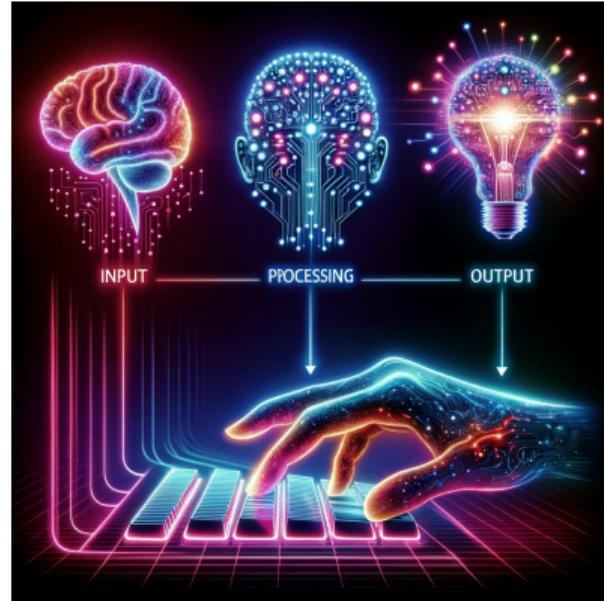


Figure: Visually stunning illustration of NLP

Rise of Large Language Models

Large Language Models represent a significant leap in AI's ability to process and generate human language, thanks to their scale and complexity.

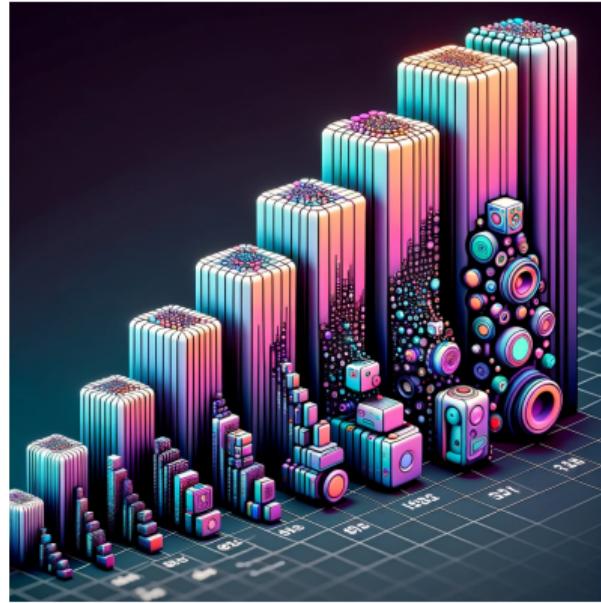


Figure: Comparison graphic of LLM against earlier models

Training Large Language Models

- ▶ Reinforcement Learning involves training models to make sequences of decisions by rewarding desired behaviors.
- ▶ In LLMs, it can be used for fine-tuning responses based on feedback.
- ▶ This in connection with human feedback is what makes ChatGPT so powerful

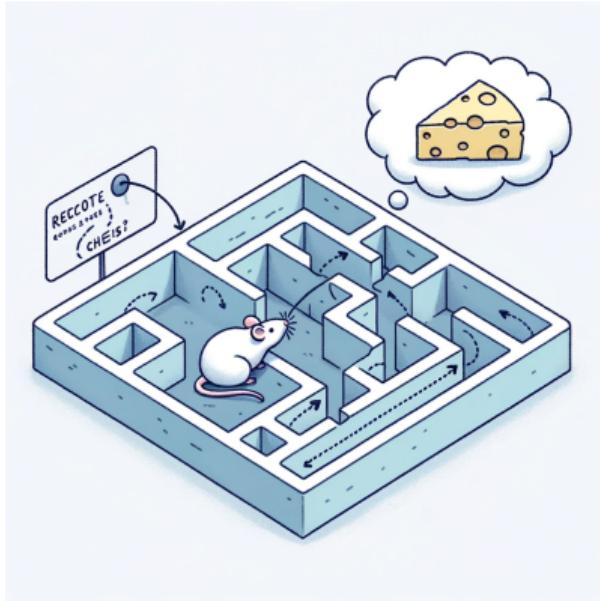


Figure: Reinforcement learning visualised as a mouse in a maze being rewarded for finding the middle

ChatGPT is Not Alone...

- ▶ GPT (OpenAI/[Microsoft](#))
- ▶ PaLM (Google)
- ▶ LLaMA (Meta)
- ▶ Gemini ([DeepMind/Google](#))
- ▶ Alpha Code (Google)
- ▶ Grok (X)

These models represent the cutting edge in natural language understanding and generation.

ChatGPT is Not Alone...

	Gemini Ultra	Gemini Pro	GPT-4	GPT-3.5	PaLM 2-L	Claude 2	Inflection-2	Grok 1	LLAMA-2
MMLU Multiple-choice questions in 57 subjects (professional & academic) (Hendrycks et al., 2021a)	90.04% CoT@32* 83.7% 5-shot	79.13% CoT@8* 71.8% 5-shot	87.29% CoT@32 (via API**) 86.4% 5-shot (reported)	70% 5-shot	78.4% 5-shot	78.5% 5-shot CoT	79.6% 5-shot	73.0% 5-shot	68.0%***
GSM8K Grade-school math (Colbe et al., 2021)	94.4% Maj1@32	86.5% Maj1@32	92.0% SFT & 5-shot CoT	57.1% 5-shot	80.0% 5-shot	88.0% 0-shot	81.4% 8-shot	62.9% 8-shot	56.8% 5-shot
MATH Math problems across 5 difficulty levels & 7 subdisciplines (Hendrycks et al., 2021b)	53.2% 4-shot	32.6% 4-shot	52.9% 4-shot (via API**) 50.3% (Zheng et al., 2023)	34.1% 4-shot (via API**)	34.4% 4-shot	—	34.8% 8-shot	23.9% 4-shot	13.5% 4-shot
BIG-Bench-Hard Subset of hard BIG-bench tasks written as CoT problems (Srivastava et al., 2022)	83.6% 3-shot	75.0% 3-shot	83.1% 3-shot (via API**)	66.6% 3-shot (via API**)	77.7% 3-shot	—	—	—	51.2% 3-shot
HumanEval Python coding tasks (Chen et al., 2021)	74.4% 0-shot (IT)	67.7% 0-shot (IT)	67.0% 0-shot (reported)	48.1% 0-shot	—	70.0% 0-shot	44.5% 0-shot	63.2% 0-shot	29.9% 0-shot
Natural2Code Python code generation. (New held-out set with no leakage on web)	74.9% 0-shot	69.6% 0-shot	73.9% 0-shot (via API**)	62.3% 0-shot (via API**)	—	—	—	—	—
DROP Reading comprehension & arithmetic. (metric: F1-score) (Dua et al., 2019)	82.4 Variable shots	74.1 Variable shots	80.9 3-shot (reported)	64.1 3-shot	82.0 Variable shots	—	—	—	—
HellaSwag (validation set) Common-sense multiple choice questions (Zellers et al., 2019)	87.8% 10-shot	84.7% 10-shot	95.3% 10-shot (reported)	85.5% 10-shot	86.8% 10-shot	—	89.0% 10-shot	—	80.0%***
WMT23 Machine translation (metric: BLEURT) (Tom et al., 2023)	74.4 1-shot (IT)	71.7 1-shot	73.8 1-shot (via API**)	—	72.7 1-shot	—	—	—	—

Figure: Comparison of LLMs (not generated by DALL-E)

ChatGPT for Developers

Leveraging ChatGPT in software development:

- ▶ Automating code generation, **testing**, and debugging.
- ▶ Integrating natural language interfaces into applications.
- ▶ Enhancing software documentation and support systems.
- ▶ Lowering the threshold for new developers to join a project

Ideal for streamlining development workflows and improving productivity.



Figure: Developer working with AI assistance

ChatGPT for Management

Empowering management with ChatGPT:

- ▶ Enhancing decision-making with data-driven insights.
- ▶ Streamlining report generation, **translation**, and documentation.
- ▶ Providing interactive tools for strategy planning and analysis.

A valuable asset for efficient and informed managerial decisions.



Figure: Board room where ChatGPT is assisting

ChatGPT for Economists

ChatGPT's role in economics:

- ▶ Conducting in-depth market analysis and forecasting.
- ▶ Assisting in economic research and data interpretation.
- ▶ Predicting trends based on large datasets.

A powerful tool for data-driven economic insights and research.



Figure: Economist creating market forecasts using ChatGPT

ChatGPT in Personal Life: Learning and Productivity

Enhancing daily life with ChatGPT:

- ▶ Learning new skills and languages.
- ▶ Personalized productivity and organization tools.
- ▶ Customized educational support and tutoring.

A personal assistant to foster continuous learning and efficiency.



Figure: People using AI for educational purposes

ChatGPT in Personal Life: Entertainment and Creativity

ChatGPT for leisure and creativity:

- ▶ Generating creative writing, music, and art ideas.
- ▶ Interactive entertainment, like games and storytelling.
- ▶ Planning travel and leisure activities.

A source of inspiration and fun in everyday life.



Figure: Family creating telling AI generated stories together

The Rapid Advancements of LLMs

- ChatGPT was released 22nd November 2022
- Significant improvements in accuracy and understanding.
- Expanded applications in various sectors.

A year of rapid growth setting the stage for future developments.

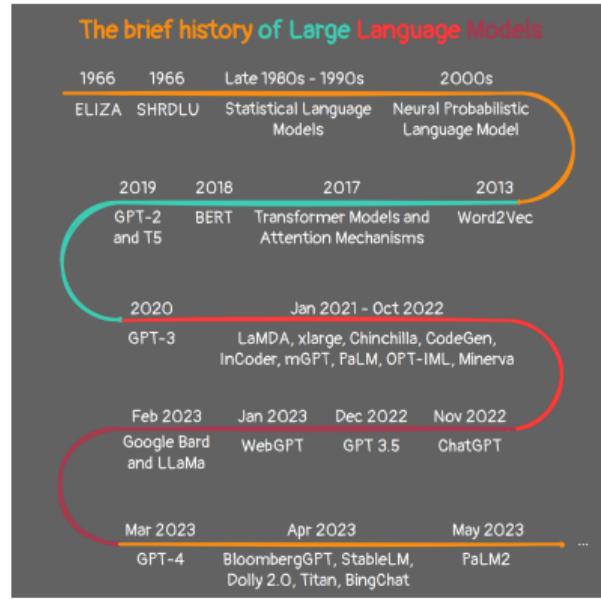


Figure: Borrowed from a Level Up Coding blog post by Armin Norouzi, Ph.D

Predictions for the Future of LLMs

Taking into account that the large language models of today are built on extremely simple principles:

- ▶ Anticipated advancements in AI capabilities and autonomy.
- ▶ Potential for transformative impacts across industries.
- ▶ Ethical and societal considerations.

Envisioning a future shaped by continuous AI innovation.



Figure: Conceptual illustration of the future of LLMs

The Importance of Awareness

- ▶ The AI models we use today are "dumb"
- ▶ "It seems probable that once the machine thinking method had started, it would not take long to outstrip our feeble powers"¹
- ▶ "AI is years away from passing the Turing test" - ChatGPT-4 already have²
- ▶ "... the bad case, I think this is important to say, it's lights out for all of us."³
- ▶ Listening suggestions:
 - ▶ "Future of AI" by Lex Fridman (MIT) and Sam Altman (CEO of OpenAI)⁴
 - ▶ "Sommarprat" by Max Tegnmark (MIT)⁵

¹ Alan Turing, 1951

² <https://www.nature.com/articles/d41586-023-02361-7>

³ Sam Altman (CEO of OpenAI)

⁴ https://youtu.be/L_Guz73e6fw?si=c8B_b29VgV_BKuP2

⁵ <https://sverigesradio.se/avsnitt/max-tegmark-sommarpratare-2023> (Swedish)

Adapting to the Future of LLMs

Is that it? Is everything just preparations for our future silicon-based overlords? Of course not!

The importance of embracing AI advancements:

- ▶ Staying ahead in a rapidly evolving technological landscape.
- ▶ Leveraging AI for competitive advantage and innovation.
- ▶ Preparing for ethical and practical challenges.

Not joining in is not just a missed opportunity but a strategic oversight.



Figure: Evolving technological landscape with metaphors

Building a ChatGPT App

Applicability Demonstration

Query:



You

Considering that I've never before written any frontend code (I've heard about React, Node, Angular, and Vue but I can't differentiate between them), which environment would you consider a good start for this project? Give a brief list of arguments

Response:



ChatGPT

Given your situation, where you're new to frontend development but have heard of React, Node.js, Angular, and Vue.js, I would recommend starting with React for this project. Here's a brief comparison and reasoning:

1. React:

• Pros:

- **Popularity and Community Support:** React is widely used and has a large community, which means plenty of resources, tutorials, and support available.
- **Component-Based Architecture:** Makes it easier to understand and manage code, especially for beginners.
- **Flexibility:** React can be integrated with various other libraries and tools, allowing for more customized development.
- **Good for Single-Page Applications:** Ideal for a project like a ChatGPT interface.

• Cons:

- **Learning Curve:** While React itself is straightforward, learning the ecosystem around it (like state management tools) can be challenging.

2. Node.js:

• Note:

- Note: Node.js is not a frontend framework; it's a JavaScript runtime for the backend. However, it's often used alongside frontend frameworks to build full-stack applications.

3. Angular:

• Pros:

- **Complete Framework:** Offers a comprehensive solution with lots of built-in functionalities.
- **Two-Way Data Binding:** Simplifies certain tasks in complex applications.

• Cons:

- **Complexity:** Generally has a steeper learning curve than React and Vue.js.
- **Verbose and Complex Syntax:** Can be overwhelming for beginners.

4. Vue.js:

• Pros:

Building a ChatGPT App

Applicability Demonstration

Query:

You
Let's go for a React framework then. What dependencies do I need to install? React is dependent on what? I'm running the OS fedora, so how would I install everything I need?

Response:

ChatGPT

Great choice going with React! To set up a React project on Fedora, you'll need to install Node.js and npm (Node Package Manager) first, as React and its dependencies are managed through npm. Here's a step-by-step guide:

1. Install Node.js and npm

Node.js is a runtime environment for JavaScript, and npm is its package manager. React requires both.

- Open your terminal.
 - To install Node.js and npm on Fedora, you can use the default package manager, `dnf`. Type the following command:

```
bash                                     ⌂ Copy code

sudo dnf install nodejs

This command will install both Node.js and npm as they come bundled together
in Fedora's repositories.
```

```
bash
```

node -v
npm -v

This will display the installed versions of Node.js and npm, respectively.

2. **Create a React App:**
Once Node.js and npm are installed, you can create a new React app using '`create-react-app`', which is an officially supported way to create single-page React applications. It sets up the development environment so you can use the latest

Building a ChatGPT App

Applicability Demonstration

Query:

You
Perfect, that worked.
Now I want to change the "src" code such that the window looks like a chat window. In other words, I want to know which file to alter and alter it such that:
* the webpack contains a window for displaying the chat history and new messages at the top 80% of the window (similarly to how the web UI for this window looks)
* the webpack shows an input box at the bottom 20% of the window where the input box is supposed to take text from a user.

Response:

"src" directory. Here's a step-by-step guide:

- 1. Modify the App Component:**
 - * Open your React project in a code editor.
 - * Go to the 'src' directory.
 - * The primary file you'll be working with is 'App.js'. This is the root component of your React app.
- 2. Setup Basic Layout:**
 - * First, let's set up a basic layout for the chat window. Replace the contents of 'App.js' with the following code:

```
javascript  Copy code

import React, { useState } from 'react';
import './App.css';

function App() {
  const [messages, setMessages] = useState([]);

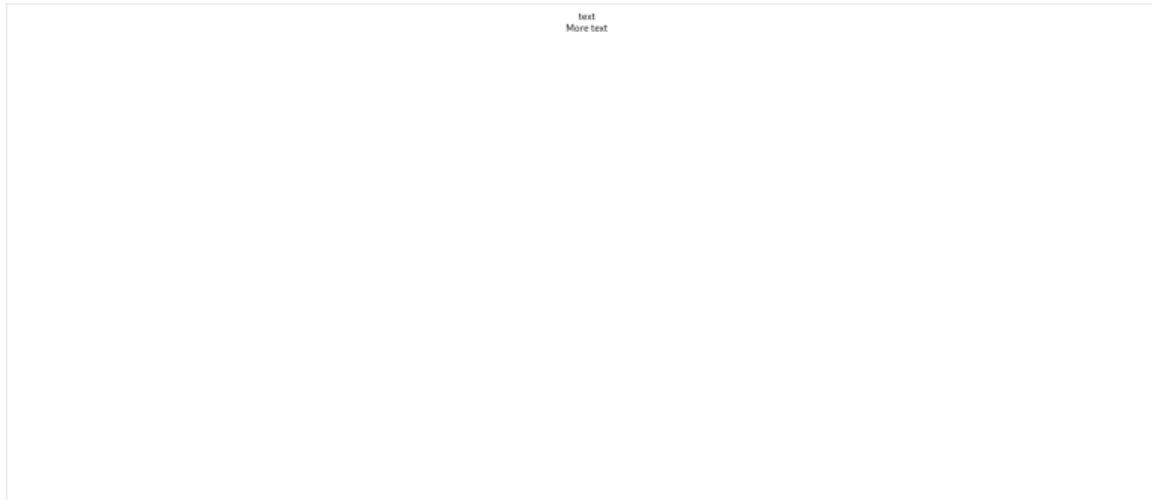
  const sendMessage = (message) => {
    if (message) {
      setMessages([...messages, message]);
    }
  };

  return (
    <div className="App">
      <div className="chat-window">
        {messages.map((msg, index) => (
          <div key={index}>{msg}</div>
        ))}
      </div>
    </div>
  );
}

export default App;
```

Building a ChatGPT App

Applicability Demonstration



Can I add Additional Text?

Send

Building a ChatGPT App

Applicability Demonstration

Query:

You
That was amazing, but could you match the background colour and text colour to the one ChatGPT uses?

Response:

ChatGPT

Sure, to match the background and text colors to those used in the ChatGPT interface, you'll need to adjust the CSS in the `'App.css'` file. The ChatGPT interface typically uses a light background with dark text. Here's how you can update the styles:

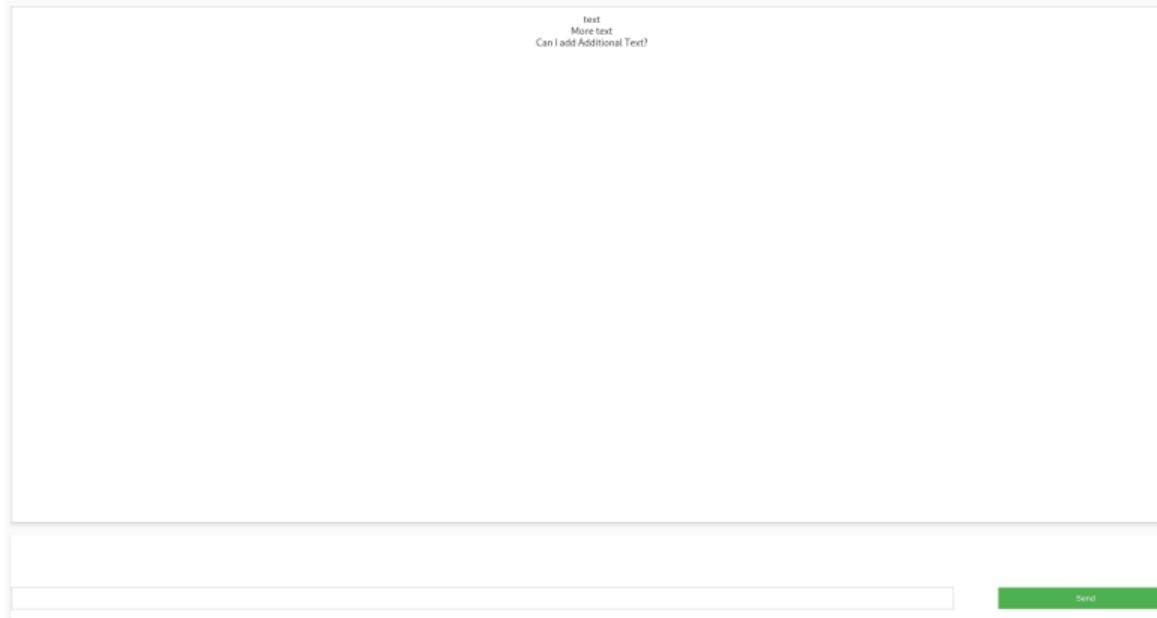
1. Modify `'App.css'`:

- * Open the `'App.css'` file in your project.
- * Replace the existing CSS with the following styles. These styles will give you a color scheme similar to ChatGPT's interface:

```
css Copy code  
body {  
    margin: 0;  
    font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto"  
    "Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue"  
    sans-serif;  
    -webkit-font-smoothing: antialiased;  
    -moz-osx-font-smoothing: grayscale;  
    background-color: #FAFAFA; /* Light background */  
    color: #333333; /* Dark text */  
}  
  
.App {  
    text-align: center;  
}
```

Building a ChatGPT App

Applicability Demonstration



Building a ChatGPT App

Applicability Demonstration

Query:

You
I like the colours and the separation; but can you make the input text from the text box be written closer to the right side of the screen?
I also want to echo the text that is written (so that it looks like I get a response). The echo is supposed to be written on the left side of the screen.

Response:

ChatGPT
To align the input text to the right and echo the messages on the left, we'll need to make some adjustments to both the JavaScript logic in '[App.js](#)' and the CSS in '[App.css](#)'. Here's how you can do it:

1. Update '[App.js](#)':
 - We need to modify the `sendMessage` function to include both the user's message and the echoed response.
 - We will also add a property to each message to distinguish between the user's message and the echoed message.

Replace your '[App.js](#)' with the following code:

```
import React, { useState } from 'react';
import './App.css';

function App() {
  const [messages, setMessages] = useState([]);

  const sendMessage = (message) => {
    if (message) {
      setMessages([...messages, { text: message, sender: 'user' }, { text: message, sender: 'echo' }]);
    }
  };

  return (
    <div className="App">
```

Building a ChatGPT App

Applicability Demonstration



Building a ChatGPT App

Applicability Demonstration

Query:

You
I like the colours and the separation; but can you make the input text from the text box be written closer to the right side of the screen?
I also want to echo the text that is written (so that it looks like I get a response). The echo is supposed to be written on the left side of the screen.

Response:



ChatGPT

To align the input text to the right and echo the messages on the left, we'll need to make some adjustments to both the JavaScript logic in `'App.js'` and the CSS in `'App.css'`. Here's how you can do it:

1. Update `'App.js'`:

- We need to modify the `'sendMessage'` function to include both the user's message and the echoed response.
- We will also add a property to each message to distinguish between the user's message and the echoed message.

Replace your `'App.js'` with the following code:

```
javascript  Copy code

import React, { useState } from 'react';
import './App.css';

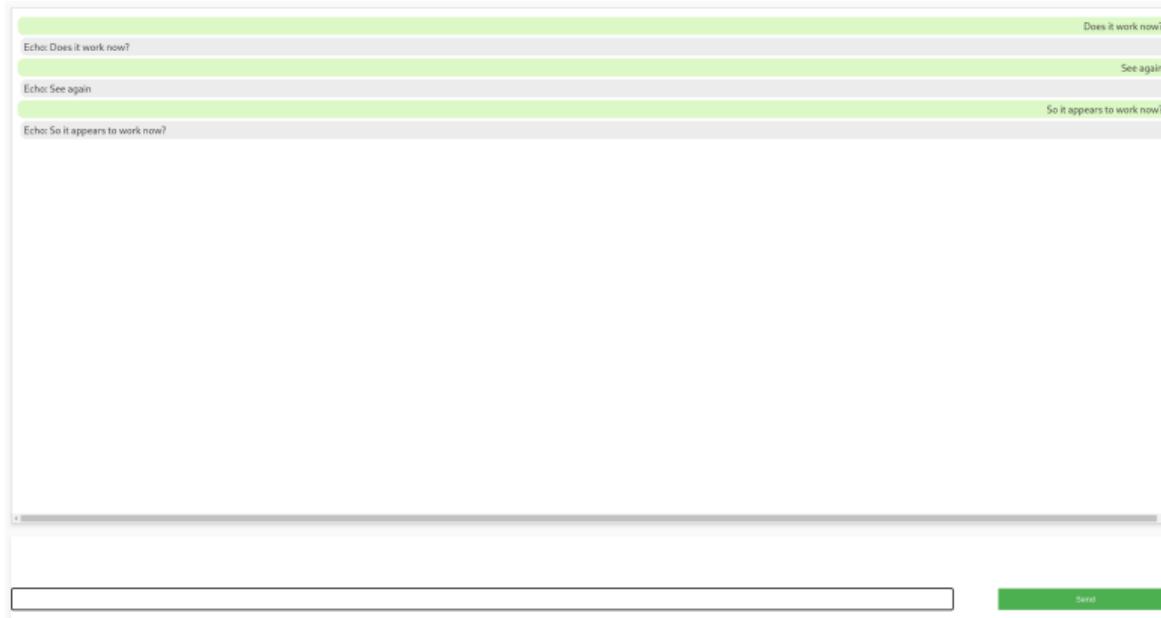
function App() {
  const [messages, setMessages] = useState([]);

  const sendMessage = (message) => {
    if (message) {
      setMessages([...messages, { text: message, sender: 'user' }, { text: message, sender: 'assistant' }]);
    }
  };

  return (
    <div className="App">
      <h1>ChatGPT</h1>
      <div>
```

Building a ChatGPT App

Applicability Demonstration



Building a ChatGPT App

Applicability Demonstration

Query:

You
That solved that problem as well.
However, I have a wide screen and the messages are quite far away from each other.
Can we make the maximum width of both the input box as well as the chat history window be smaller, say for instance the same width as the ChatGPT window?

Response:

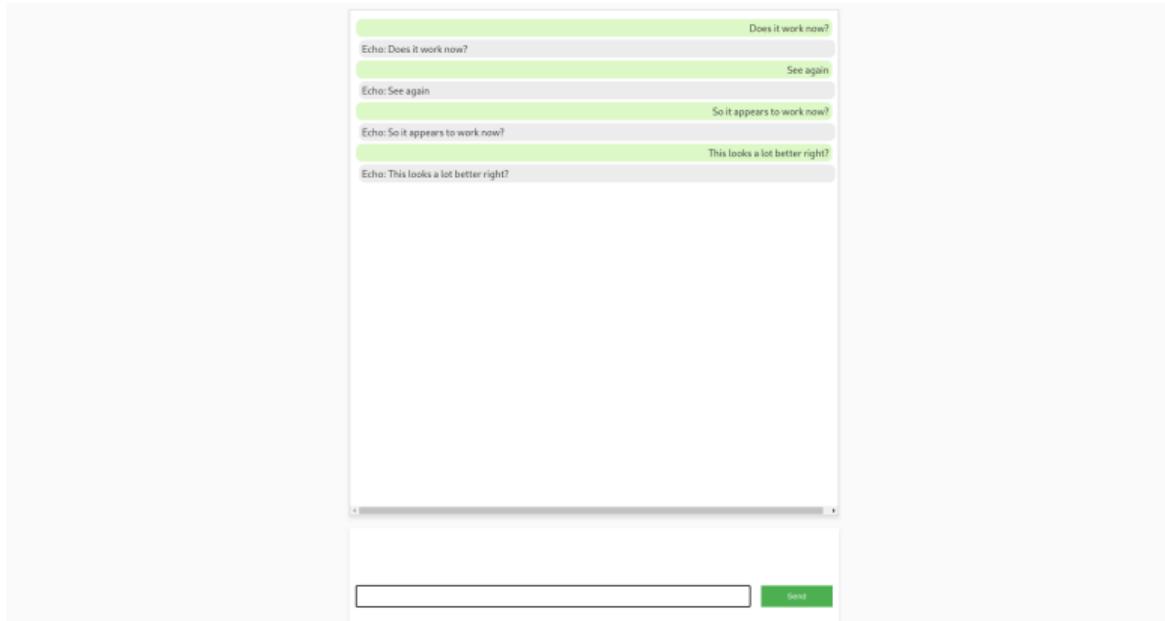
ChatGPT
Certainly! To control the maximum width of both the chat history window and the input box to better suit a wide screen, you can set a 'max-width' property and use 'margin: auto' to center them. This will make these elements visually similar to the ChatGPT interface in terms of width.

Update your 'App.css' with the following changes:

```
css Copy code  
.App {  
  text-align: center;  
  display: flex;  
  flex-direction: column;  
  height: 100vh;  
  align-items: center; /* Center the child elements */  
}  
  
.chat-window, .input-box {  
  max-width: 800px; /* Set a maximum width */  
  width: 100%; /* Use the full width up to the maximum */  
  margin: 10px auto; /* Center the elements horizontally */  
  box-sizing: border-box; /* Include padding and border in the element */  
}
```

Building a ChatGPT App

Applicability Demonstration



Building a ChatGPT App

Applicability Demonstration

Query:



You

That worked. However, I now noticed something new. When inputting a very long message into the input-box, it turns out that I can't see all text because it keeps scrolling to the right (along with my increasing message length). Can I make sure that when we reach the end of the input-box there is a line break and the input-box's input area is scaled vertically to the new text (with a maximum height of the input-box).

Response:



ChatGPT

To handle long input messages in the input box gracefully, you'll want to use a `'textarea'` instead of an `'input'` element. A `'textarea'` can easily handle multi-line input and automatically grows vertically as more text is entered. We can also set a maximum height to ensure it doesn't grow indefinitely.

Here's how to update your `'InputBox'` component in `'App.js'`:

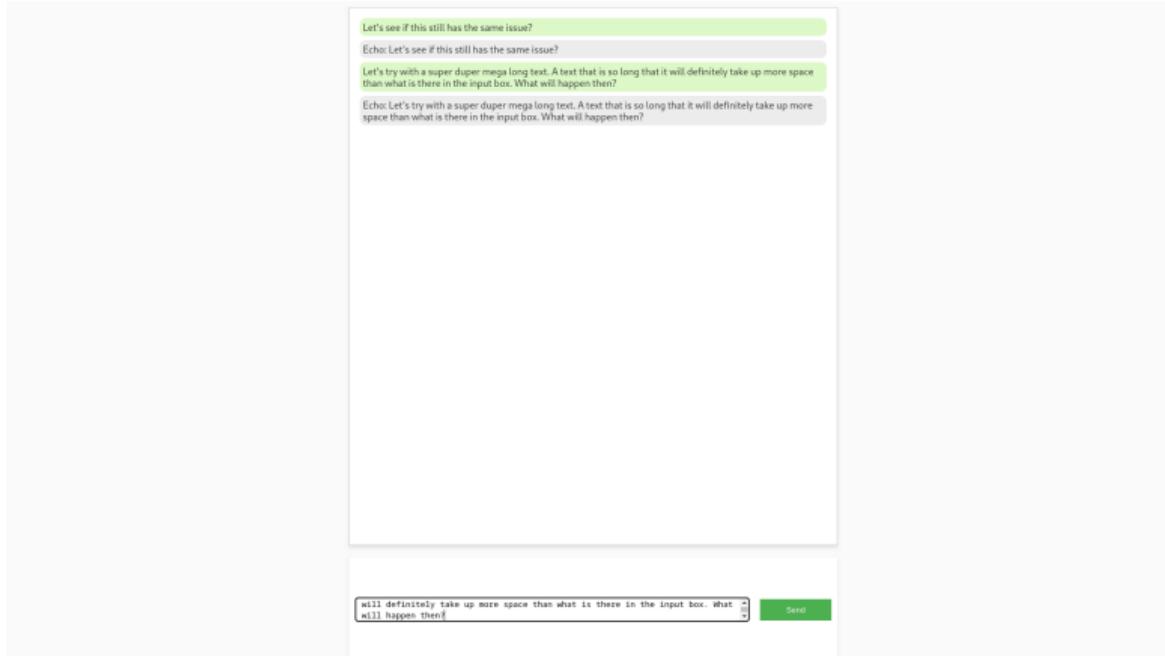
1. Update the `'InputBox'` Component:

Replace the `'input'` element with a `'textarea'` and apply some additional properties to handle its resizing.

```
javascript Copy code
function InputBox({ onSendMessage }) {
  const [input, setInput] = useState('');
  const handleSend = () => {
    onSendMessage(input.trim());
    setInput('');
  };
  const handleKeyDown = (event) => {
    if (event.key === 'Enter' && !event.shiftKey) {
      event.preventDefault();
      handleSend();
    }
  };
}
```

Building a ChatGPT App

Applicability Demonstration



Building a ChatGPT App

Applicability Demonstration⁶

The screenshot shows a messaging interface with several messages exchanged between two users:

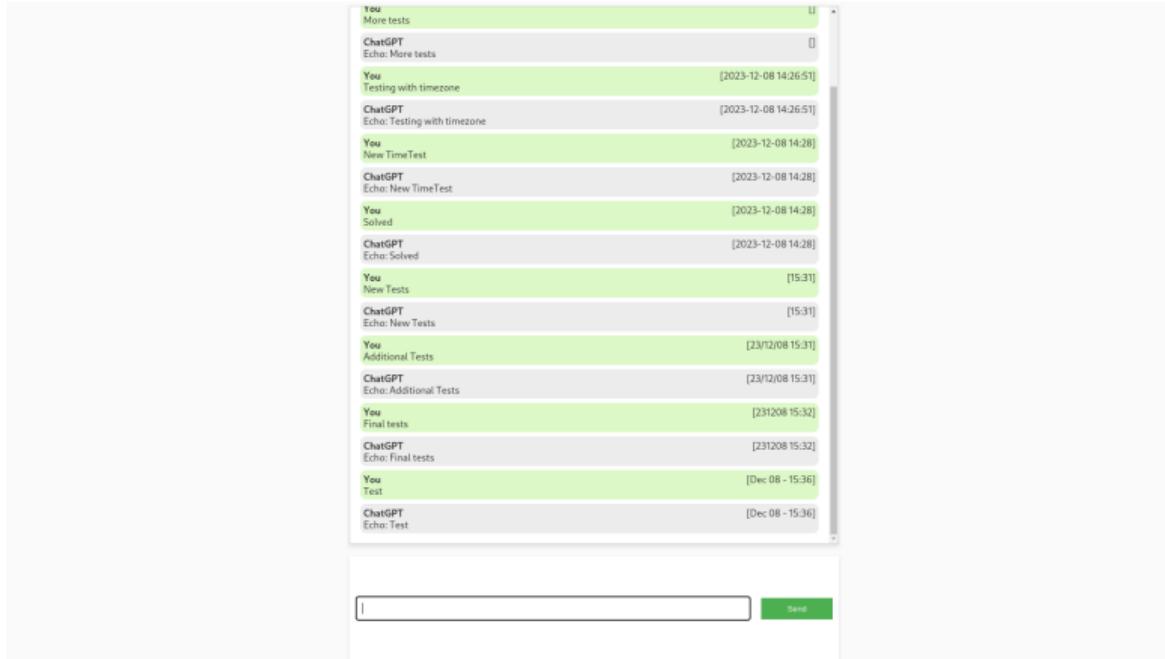
- Echo: Let's see if this still has the same issue?
- Echo: Let's see if this still has the same issue?
- Echo: Let's try with a super duper mega long text. A text that is so long that it will definitely take up more space than what is there in the input box. What will happen then?
- Echo: Let's try with a super duper mega long text. A text that is so long that it will definitely take up more space than what is there in the input box. What will happen then?
- asdf
- Echo: asdf
- Long text message (redacted)

This message is extremely long, consisting of multiple paragraphs of Latin text. It is displayed in a large, monospaced font and occupies most of the screen area.
- Send

⁶I'm gonna stop boring you with the Query ↔ Response slides now

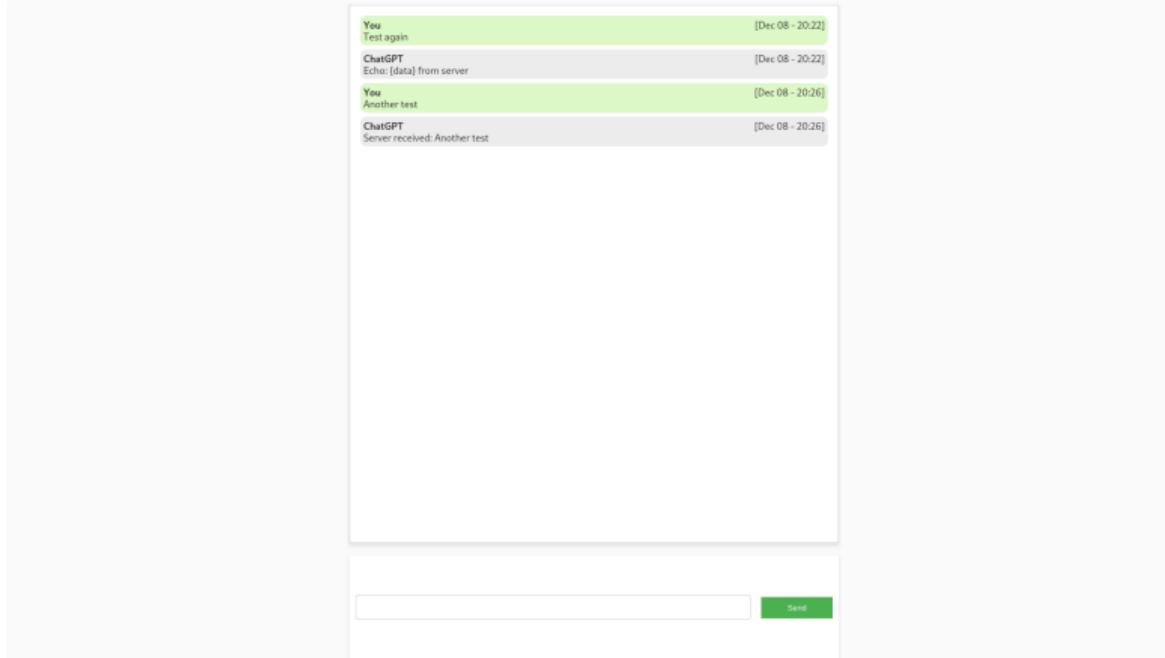
Building a ChatGPT App

Applicability Demonstration



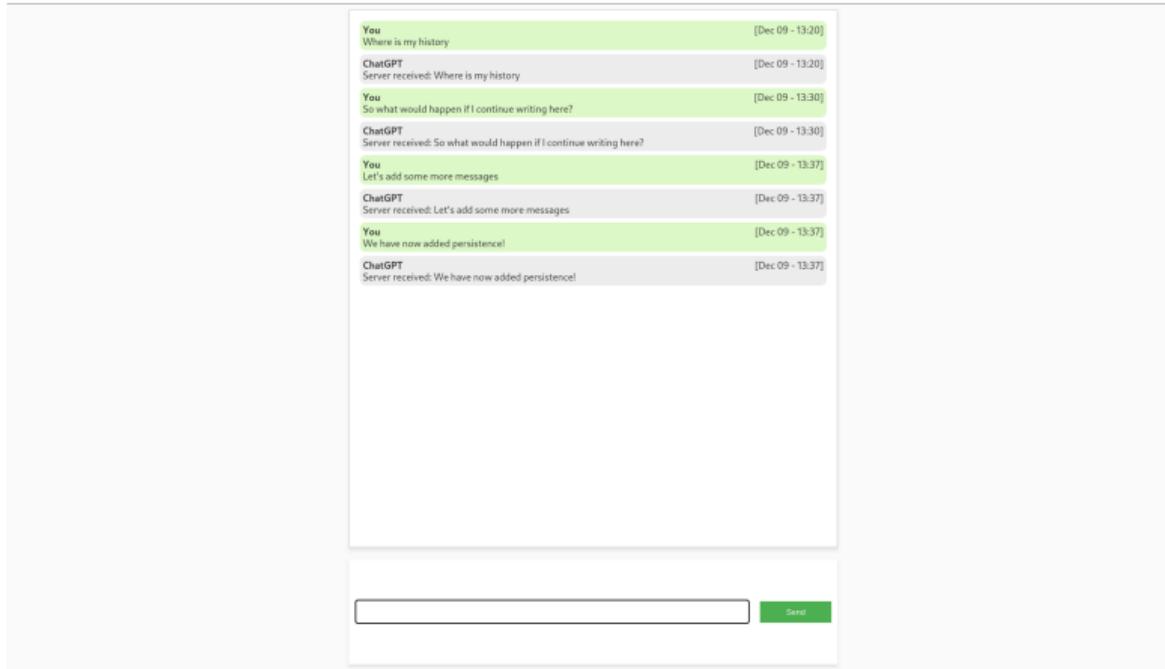
Building a ChatGPT App

Applicability Demonstration



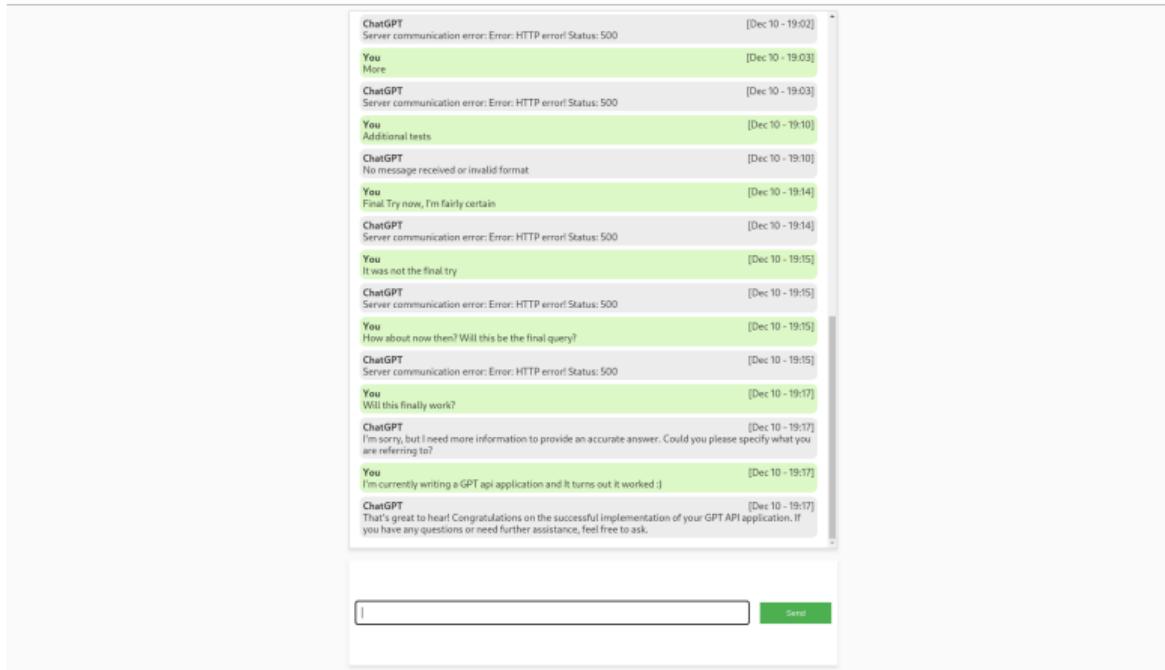
Building a ChatGPT App

Applicability Demonstration



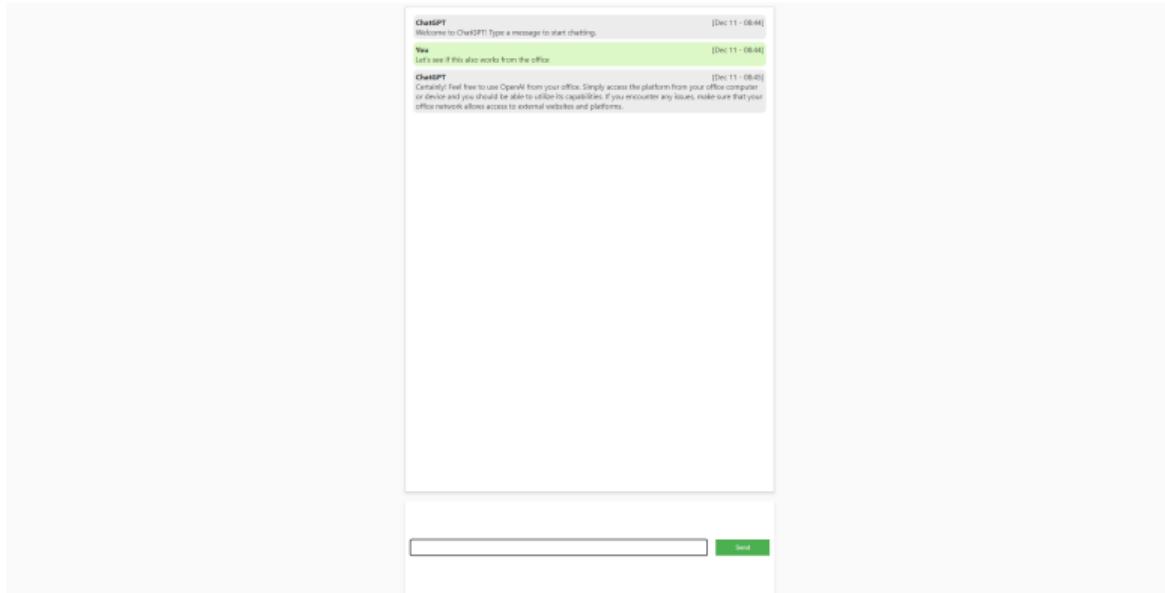
Building a ChatGPT App

Applicability Demonstration



Building a ChatGPT App

Applicability Demonstration



Building a ChatGPT App

Conclusion

If you want to verify the app generation, you can find the conversation here:

[https://chat.openai.com/share/
1caa7712-5d71-4465-9951-d1ef0adb58f5](https://chat.openai.com/share/1caa7712-5d71-4465-9951-d1ef0adb58f5)

And the repository here:

<https://github.com/NilsVreman/chatgpt-interface>

Conclusion

- ▶ Large Language Models are not yet silver bullets
 - ▶ However, models like ChatGPT represent a significant leap in AI capabilities
- ▶ Enhancing productivity professionally and enriching personal life experiences
- ▶ Staying informed and adaptive to these advancements is crucial

Thank you for your attention!^a

^aThe presentation part is finally over. Let the fun commence!



Figure: Artistic representation of AI ↔ human collaboration

Engaging Projects

- ▶ The project shouldn't take more than ~1-2 hours
- ▶ In groups of 2-3, chose a project from the list *or* come up with your own
- ▶ The goal is to get a feel for the capabilities of ChatGPT
 - ▶ Really try to maximize the ChatGPT usage
- ▶ Reconvene in 1 hour to give a 1-2 min discussion about your project and pros/cons using ChatGPT

Project Suggestions

- ▶ **Expert Knowledge Test:** Compare ChatGPT's responses on a specialized topic with an actual expert's input
- ▶ **Learn a New Programming Language:** Use ChatGPT to guide you through a basic project in a, to you, unbeknownst programming language
- ▶ **Language Translation Relay:** Translate a paragraph through multiple languages and back to the original to see how well ChatGPT retains the original meaning.
- ▶ **Science Fiction Storytelling:** Collaboratively write a short science fiction story with ChatGPT
- ▶ **AI Marketing Guru:** Create a mock product and ask ChatGPT to develop a marketing strategy, comparing it with real-world strategies
- ▶ **Code Debugging Challenge:** Use ChatGPT to debug a piece of intentionally buggy code in a familiar language

Project Suggestions

- ▶ **Master Thesis Generation:** Use ChatGPT to generate master thesis suggestions suited for your application/domain
- ▶ **Create a Classic Game:** Choose a classic 8-bit game (e.g., Pong or Snake) and generate the code for it using ChatGPT
- ▶ **Business Problem-Solving:** Present ChatGPT with a common business problem and analyze its proposed solutions for feasibility and creativity
- ▶ **Personalized Psychologist:** Use ChatGPT as a psychologist and discuss a fake problem, evaluate its advice
- ▶ **Mock Job Interview:** Conduct a mock job interview with ChatGPT as the interviewer, assessing how well it can find the best candidate
- ▶ **CV Analysis:** Analyze a CV with ChatGPT and evaluate its understanding of the candidate's skills and experience

Project Suggestions

- ▶ **Automated Budget Analysis:** Use ChatGPT to analyze a sample company budget and provide insights or identify potential cost-saving areas
- ▶ **Improve the Chatbot:** Add a simple feature to the ChatBot Application
- ▶ **Financial Investment Simulation:** Create a scenario of managing an investment portfolio, ask ChatGPT for investment advice based on current market conditions, and discuss its investment strategy
- ▶ **HR Scenario Simulation:** Run through various human resources scenarios (e.g., employee conflict) and assess ChatGPT's advice
- ▶ **Administrative Task Automation:** Create a script to automate a common administrative task
- ▶ **Building a Simple API:** Use ChatGPT to guide you through building a RESTful API for a basic application

Extra Slides

Transformers - The Core of LLMs

- ▶ Transformers are a type of model architecture that's particularly effective for processing sequential data like text.
 - ▶ The use of "**self-attention**" allows transformers to learn the relationships between words in a sentence.

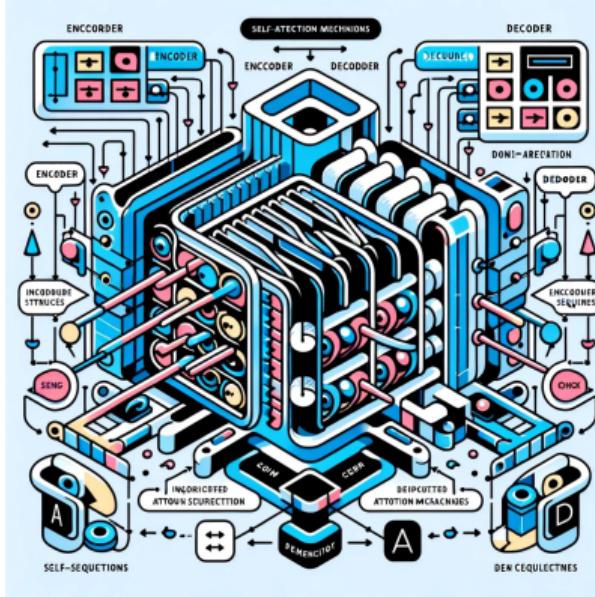


Figure: A simple representation of the encoder-decoder architecture of a transformer (This one physically hurt to allow)

Tokenization - Turning Text into Data

Tokenization is the process of converting text into smaller units (tokens) that can be easily processed by LLMs.

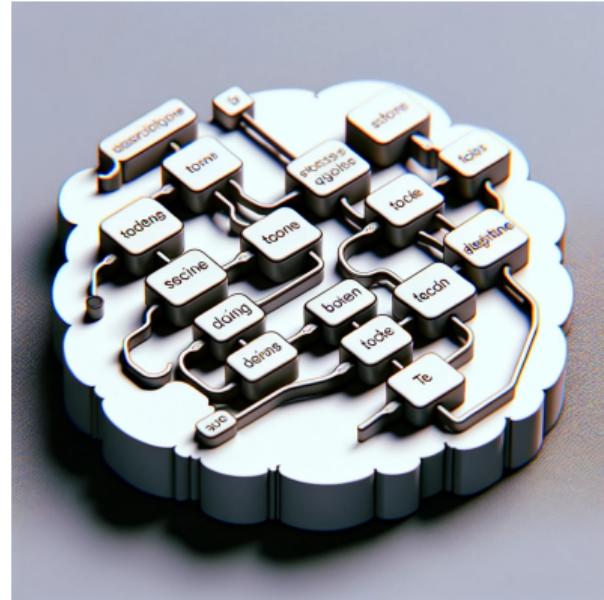


Figure: A sentence being broken down into tokens (words or subwords), maybe as a flow-diagram

Embeddings - Understanding Word Meaning

Embeddings are a way to represent tokens as vectors in a high-dimensional space, capturing their meaning and relationships.

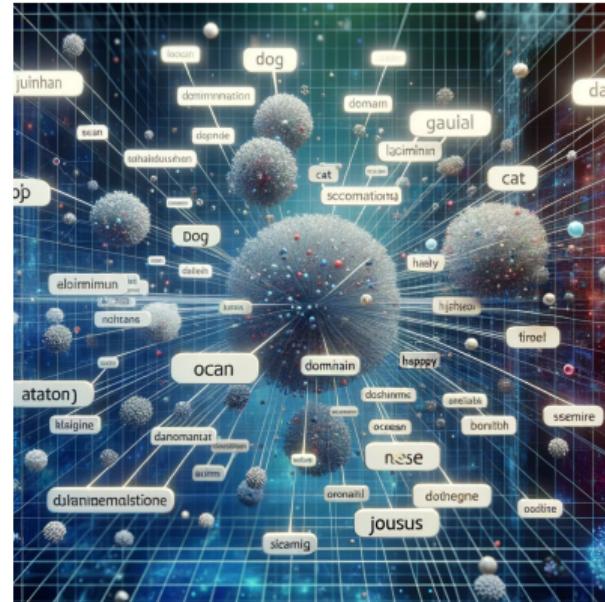


Figure: A 3D visualisation showing similar words being close together