Introduction to Generative AI and Large Language Models (LLMs)

Creating Virtual Agents using LLMs - Session 1

Monday, 28/10/2024

Presented by Dries Van Hansewijck

Dries Van Hansewijck

- Software developer
- MSc in IT Governance
- Owner of Van Hansewijck Development
  - Software consultancy & development
- https://vanhansewijck.com
- https://www.linkedin.com/in/driesvanhansewijck/

Focus of this course

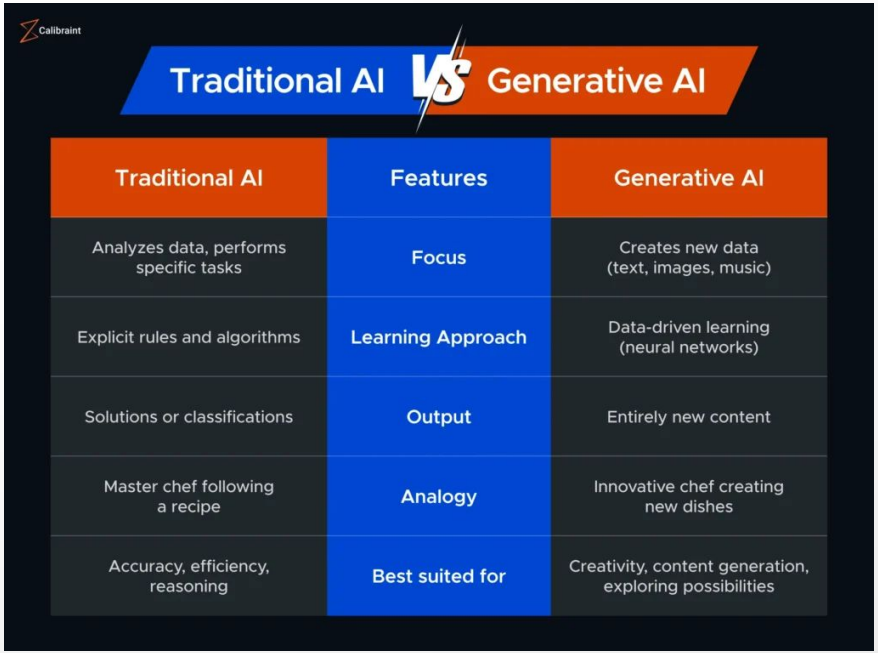
- Focus on the practical application of LLMs in the context of virtual agents
  - i.e. you are a computer engineer tasked with solving a business need
- We start with basic LLM vocabulary and a brief history
- We discuss emerging trends and continue with prompt engineering
- Finally, we introduce the assignment for this course.
- Syllabus: https://vanhansewijck.com/courses/llm-agents/intro


Introduction

What is Generative AI?

Definition: AI focused on creating new content, responses, or insights.

Contrast with Traditional AI: Generative vs. rule-based and task-specific models.



Traditional AI	Features	Generative AI
Analyzes data, performs specific tasks	Focus	Creates new data (text, images, music)
Explicit rules and algorithms	Learning Approach	Data-driven learning (neural networks)
Solutions or classifications	Output	Entirely new content
Master chef following a recipe	Analogy	Innovative chef creating new dishes
Accuracy, efficiency, reasoning	Best suited for	Creativity, content generation, exploring possibilities

https://www.calibraint.com/blog/generative-ai-vs-traditional-ai

Image upscaling example

- Minecraft wallpaper

Understanding Large Language Models (LLMs)

Definition of LLMs: Advanced neural networks trained on large language datasets.

Abilities: Interpret complex language, generate human-like responses, adapt to different domains.

https://www.wisecube.ai/

Real-World Applications of LLMs

Brief descriptions of practical applications in the context of virtual agents:

1. Virtual Assistant for Booking Websites
2. Company Policy Assistant
3. Research Assistant for Students
4. Project Appraisal Assistant

Example: Virtual Assistant for a Booking Website

Goal: Provide real-time assistance on booking availability, pricing, and cancellations to end-users. (external)

Benefits:

- Enhances user experience, reduces customer service workload.
- The user can interact with the agent in a natural language style conversation.
- Virtual agents are 24/7 online.

Example: Company Policy Assistant

Goal: Provide assistance to employees on company policies. (internal)

Problem: Modern companies have a huge database of (internal) documents and policies. It is not always easy for employees to find the relevant policies and documents for their specific use case.

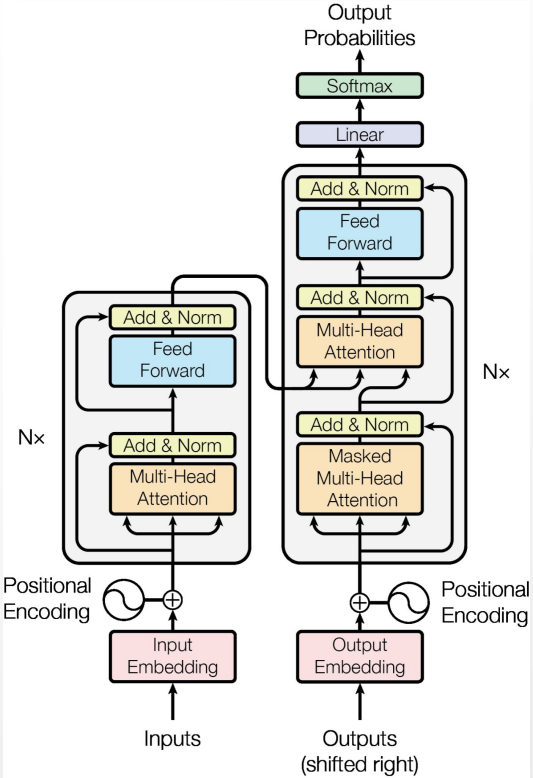
Eg: what is the policy around a data breach?

LLMs can help with finding the right document and answering employee questions through the use of RAG (will be discussed next week)

Benefits: Enhances user experience, reduces time searching => **increased productivity**

Key Components of LLMs

- **Training Data:** Sourced from books, websites, etc.
- **Transformer Architecture:** Multi-layered model structure with attention mechanisms.
- **Fine-Tuning & Prompting:** Adjusts model behavior for specific tasks.



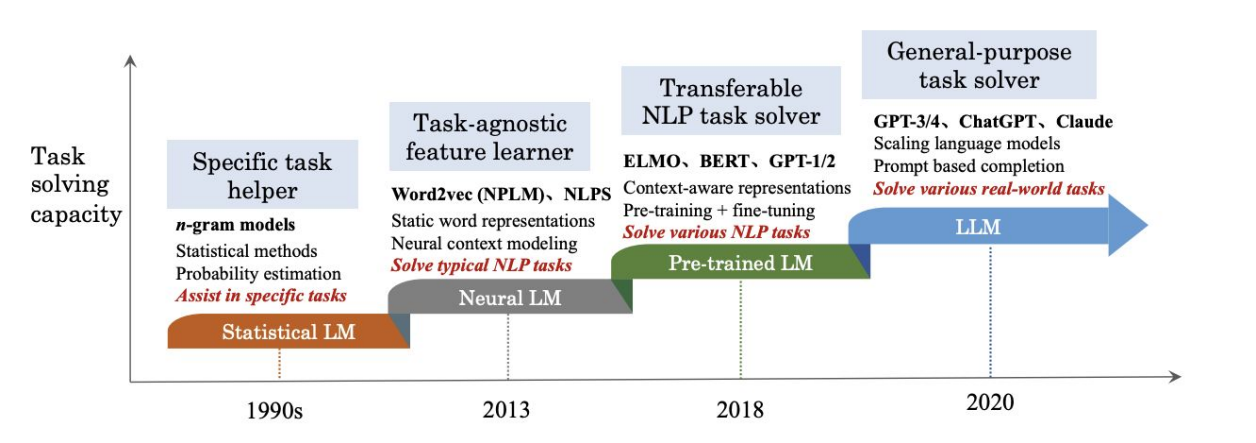
https://www.run.ai/

LLMs in the Evolution of AI

Adaptability: LLMs can handle diverse tasks with minimal customization.

Accessibility: Interaction through natural language.

Scalability: Easy to implement in various industries.



https://klu.ai/glossary/large-language-model

Key Takeaways

- Generative AI allows AI to create new, contextual responses.
- LLMs' architecture enables nuanced and complex interactions.
- Practical applications span multiple domains, highlighting adaptability.


History

Key Milestones in LLM Development

1950s - 2000s: Early foundations with neural networks and statistical language models.

2010 - 2015: Rise of neural networks in NLP, with models like **Word2Vec** and **seq2seq**.

2018 - 2020: Breakthroughs with **Transformers** (BERT, GPT-2, GPT-3).

2021 - Present: Specialization and efficiency focus (e.g., **T5** and responsible AI practices).

Significance of LLMs in AI Development

Scalability: LLMs can handle multiple NLP tasks without specific retraining.

Human-Machine Interaction: Enhanced communication with natural language understanding.

Versatility: Adaptable across industries, from customer service to education.


Providers

LLM providers vs LLM implementors

- LLM providers develop the core LLM technology (including Transformer architecture) and train the LLMs on vast amounts of data (billions of tokens). The output is called a base model.
  They regularly roll out new versions of their models (eg OpenAI GPT3.5 => GPT4) and provide different versions (eg OpenAI GPT4o vs GPT4o-mini) for different use cases.
- LLM implementors use these base models in their applications.

Major LLM Providers

- **OpenAI**
- **Google AI**
- **Meta**
- **Anthropic**
- **Cohere**

OpenAI

Key Models: GPT-3, GPT-4

Specialization: Versatile, large-scale models ideal for language generation, conversation, and API integration.

Notable Applications: Virtual assistants, customer support.

Strengths: High performance and accessibility via APIs.

Google AI

Key Models: BERT, T5, PaLM

Specialization: Search optimization, language comprehension, few-shot learning.

Notable Applications: Google Search, Assistant, translation.

Strengths: Strong in search and language comprehension.

Meta AI

Key Models: LLaMA, BlenderBot

Specialization: Open-source models, focus on conversational AI.

Notable Applications: Social media chatbots, community engagement.

Strengths: Open access and research collaboration.

Anthropic

Key Models: Claude series

Specialization: Safe, ethical AI with controllability focus.

Notable Applications: Regulated industries, customer support.

Strengths: Emphasis on safety, ethical standards, and compliance.

Cohere

Key Models: Command series

Specialization: Natural language understanding, efficient in text classification and semantic search.

Notable Applications: E-commerce search, content analysis.

Strengths: Developer-friendly, highly efficient for NLP tasks.


Emerging
trends

Emerging Trends in LLM Development

Recent advancements shaping the future of LLMs:

- **Efficiency and scaling**
- **Multilingual capabilities**
- **Domain-specific adaptations**
- **Ethics and responsible AI**
- **Human-AI collaboration**

Trend 1: Model Efficiency and Scaling

Objective: Improve resource efficiency and scalability.

Techniques: Parameter efficiency, model distillation, sparsity, and quantization.

Applications: More accessible LLMs for virtual agents, especially on mobile platforms.

Trend 2: Multilingual Capabilities

Objective: Enable LLMs to support multiple languages.

Techniques: Diverse language datasets, cross-lingual transfer learning.

Applications: Global customer support, multilingual virtual assistants.

Trend 3: Domain-Specific Adaptations

Objective: Tailor LLMs for specialized industries like healthcare and finance.

Techniques: Domain fine-tuning, hybrid models.

Applications: Health and legal virtual assistants, regulatory compliance tools.

Trend 4: Ethics and Responsible AI

Objective: Promote safe, ethical use of LLMs.

Key Areas: Bias mitigation, content moderation, transparency.

Applications: Trustworthy virtual agents in education, customer service.

Trend 5: Human-AI Collaboration

Objective: Enhance productivity by integrating human and AI inputs.

Techniques: Interactive AI systems, feedback loops.

Applications: Collaborative content creation, real-time assistance.

Visual: Diagram showing human-AI collaborative workflow.

Impact on Virtual Agents

Trends enhance virtual agents:

- **Efficiency boosts accessibility.**
- **Multilingual models reach diverse audiences.**
- **Ethical practices foster user trust.**
- **Collaboration empowers productivity.**

Case: OpenAI Realtime API & Twilio

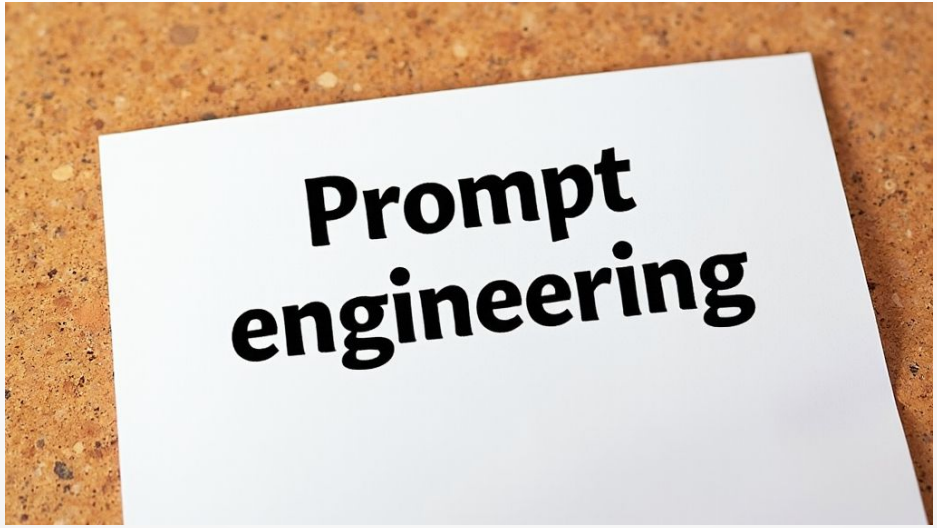
Combines LLMs and VOIP providers

- **The Realtime API supports low-latency speech-to-speech interactions for conversational AI experiences.**
- **It integrates audio input/output in the Chat Completions API, handling end-to-end audio processing in one call.**
- **Use cases include language learning, customer support, and more personalized AI interactions.**
- **Pricing is based on text and audio tokens, with details available for cost calculation.**
- **Safety and privacy are emphasized, with multiple protections against misuse.**

More info:

https://openai.com/index/introducing-the-realtime-api/

https://www.twilio.com/en-us/blog/twilio-openai-realtime-api-launch-integration


Prompt
engineering

What is a Prompt?

Definition: Input text guiding LLM responses.

Types of Prompts:

- **User Prompt:** Direct input from the user.
- **System Prompt:** Pre-set instructions shaping model behavior.

Criteria	User prompts	System prompts
Purpose	Task-specific instructions	Overall framework & guidelines
Frequency of use	Used frequently, often changed	Set once, rarely changed
Scope	Narrow, focused on individual tasks	Broad, applies to all interactions
Content focus	Specific details, context & desired outcomes	General rules, tone, ethics & brand values
Example	"Write a follow-up email to prospect X about Y product"	"You are a seasoned account executive for a B2B SaaS company..."
Typical length	Short to medium (1-5 sentences)	Medium to long (paragraph or more)
Primary impact	Output content & structure	Overall tone, behavior & approach
When to use	For each specific task or request	At the beginning of AI solution setup or a new session
Modifiability	Easily modified for each new task	Requires careful consideration to change

regie.ai

System prompts

Definition: Pre-set instructions for consistent tone and behavior.

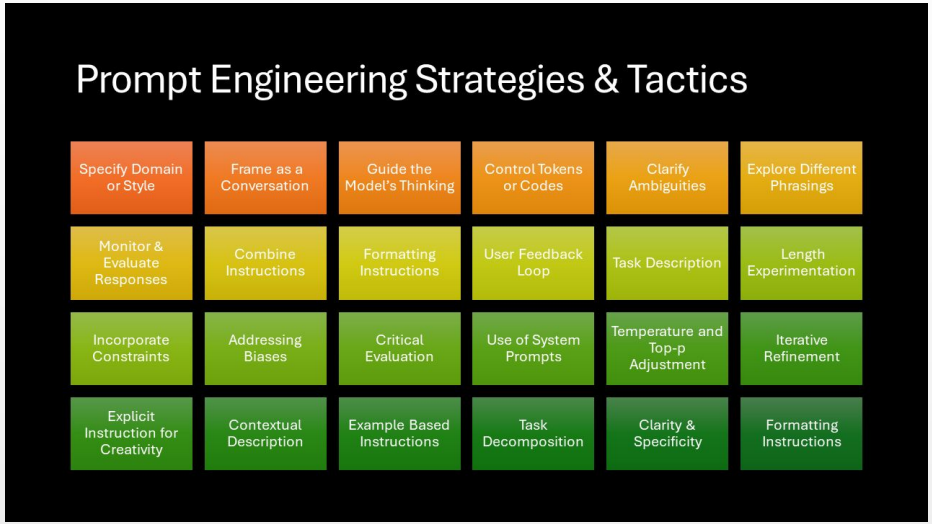
Use Cases:

- **Professional Assistant** (formal language)
- **Friendly Assistant** (informal language)
- **Industry-Specific** (e.g., healthcare, finance).

What is Prompt Engineering?

Practice of designing prompts for desired outputs.

Objective: Tailor prompts for relevance, clarity, and tone.



https://www.linkedin.com/pulse/mastering-prompt-engineering-strategies-tactics-krishna-srikanth-k-f8enc/

Key Principles for Effective Prompts

Clarity and Specificity: Make prompts clear and concise.

Context Provision: Include background information.

Iterative Refinement: Adjust prompts to improve responses.

Practical Prompting Examples

Simple Prompt: “Summarize this text.”

Enhanced Prompt: “Summarize this research on climate change for a general audience.”

Role-Playing Example: “You are a customer support assistant for a booking platform. Answer questions on cancellations and refunds.”

See https://www.promptingguide.ai/ for a list of prompting techniques

Prompt Engineering Workflow

Step-by-step process:

1. **Basic task prompt.**
2. **Add context and role.**
3. **Specify tone and formality.**
4. **Refine based on output.**

Demo conversation

- Minecraft buddy
- ChatGPT
  - Custom GPT

Key Takeaways for Effective Prompting

- System prompts set foundational behavior.
- Prompt engineering helps tailor responses to specific needs.
- Iterative refinement improves output quality.


Assignment

Assignment overview

- **Objective:** Design a virtual assistant in a chosen domain and use case.
- **Goal:** Apply LLM knowledge to develop a functional assistant.
- **Key Requirements:** Select a domain, define a use case, create prompts, develop and test.
- **Deliverables:** Report and presentation with video.

Step 1: Domain and Use Case Selection

Suggested Domains: Banking, Education, Healthcare, Real Estate, Project Management.

Example Use Cases:

- Customer Support Assistant for an online store.
- Research Helper for students.
- HR Policy Guide for employees.
- Project Appraisal Form Assistant.

Step 2: Persona and Audience

- **Define Persona:** Traits like friendliness, professionalism, formality.
- **Example Personas:**
  - Customer Support: Friendly and conversational.
  - Research Helper: Informative and concise.
  - Project Appraisal Assistant: Step-by-step guidance.
- **Target Audience:** Identify who the assistant will serve and their needs.
- **Visual:** Sample persona profiles with characteristics.

Step 3: Prompt Design

- **System Prompt:** Define the assistant's role and tone.
  - Example: "You are a friendly customer support assistant…"

- **User Prompts:** Example questions guiding typical responses.
  - "How can I check the status of my order?"
  - "What's the return policy for electronics?"

- **Refinement:** Experiment and adjust prompts to improve responses.

Assignment Deliverables

- **Report:**
  - Domain and use case explanation.
  - Approach, prompt techniques, and data sources.
  - Final system prompt.
- **Presentation (Video):**
  - 5–10 minutes explaining the assistant's purpose, persona, and sample conversations.
  - **Sample Interactions:** Demonstrate "Happy Path" and "Jailbreak Attempt."

Tips for Success

- **Choose Unique Use Cases:** Be creative with domain selection.
- **Audience Focus:** Tailor responses to audience needs.
- **Iterate on Prompts:** Experiment to refine assistant behavior.


Next
week

Next week

- LLM playground
- Prompting part 2
- RAG
- Practical Exercises


FINALLY

Outro

Thank you for your attention.

Please fill in the Google Form:
https://forms.gle/bCeT1bm24YqiADAx8

Reach out to me if you have any questions or are interested in further information.