



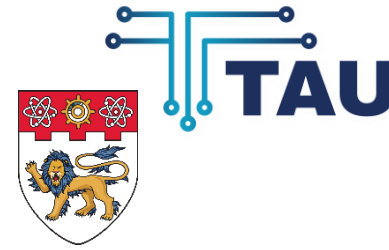
Netherlands Forensic Institute  
*Ministry of Justice and Security*



# Large Language Models: Prompt Engineering and Retrieval Augmented Generation for Digital Forensics

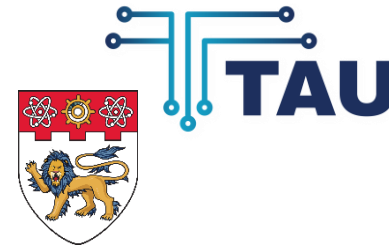
Hans Henseler (NFI, UoSL)  
Kwok-Yan Lam (NTU)  
Zee Kin Yeong (SAL)  
Victor C.W. Cheng (TauExpress)

DFRWS APAC Workshop,  
October 17, 2023, Singapore



# Agenda

| Time  | Title  |
|-------|--|
| 13:00 | Part I: Introduction Large Language Models                 |
| 13:45 | Part II: Hands-on prompt engineering for digital forensics |
| 14:30 | Part III: Hands-on with a LLM in a Google Colab notebook   |
| 15:00 | Break  |
| 15:30 | Part III continued   |
| 16:00 | Part IV: Hands-on with Retrieval Augmented Generation      |
| 16:30 | Panel discussion on LLMs in the legal domain               |
| 17:00 | End  |



# Agenda

| Time  | Title  |
|-------|--|
| 13:00 | Part I: Introduction Large Language Models                 |
| 13:45 | Part II: Hands-on prompt engineering for digital forensics |
| 14:30 | Part III: Hands-on with a LLM in a Google Colab notebook   |
| 15:00 | Break  |
| 15:30 | Part III continued   |
| 16:00 | Part IV: Hands-on with Retrieval Augmented Generation      |
| 16:30 | Panel discussion on LLMs in the legal domain               |
| 17:00 | End  |



Netherlands Forensic Institute  
*Ministry of Justice and Security*



# Introduction Large Language Models

Part I: Hans Henseler



# Microsoft Copilot

- > 2021 GitHub Copilot
- > February 1: Bing Chat
- > September 26 : Windows 11 Copilot
- > November 1: Microsoft Office 365 Copilot:



[Introducing Microsoft 365 Copilot | Your Copilot for Work - YouTube](#)





# The rise of deep learning 2012-2022

- 2012:** AlexNet wins the ImageNet Large Scale Visual Recognition Challenge
- 2014:** Introduction of Generative Adversarial Networks (GAN's)
- 2015:** AlphaGo defeats world champion Go, Lee Sedol
- 2017:** Google introduces BERT improving ML translations
- 2018-2021:** Introduction of GPT-2, DALL-E, CLIP, GPT-3, ...
- 2022:** DALL-E2, Midjourney, Stable Diffusion, ChatGPT, ...





# What is ChatGPT?

## > **ChatGPT is a large language model (LLM)**

- Essentially a machine learning model that learns *an algorithm* to predict the next word based on many text examples

## > **Based on GPT3.5/GPT4 (Generative Pre-trained Transformer)**

- Improved version of GPT-3 that “understands” text and program code
- Different models for performance, chat, text and code completion
- GPT3.5 was trained on 570 GB data from the internet (articles, posts, web pages and books)

## > **Available as**

- Free version
- ChatGPT-plus €23 per month
- OpenAI playground (API access):
  - GPT3.5-turbo API 0,002 dollar per 1.000 tokens, ~700 words
  - GPT4 API 0,03 dollar per 1.000 tokens, ~700 words



# What can ChatGPT do?

Chat. Like a chatbot that...

- > Assists with writing and brainstorming
- > Tells riddles, jokes, stories
- > Plays games
- > Gives compliments and advise
- > Helps with filling in forms

But it that can also:

- > Summarise
- > Translate
- > Analyse and structure (unstructured) information
- > Answer questions (but the answer may not be right)
- > Assist with software writing and debugging
- > Generate (anonymous) testdata





# What can ChatGPT not do?

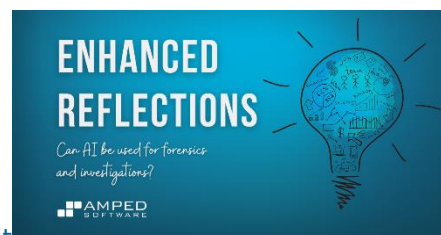
- > It hallucinates facts
- > It gives wrong answers
- > Replies can be biased
- > Can not act spontaneously (needs to be prompted)
- > Is not good at making calculations (e.g.  $4213 \times 8242$ )
- > Is limited to generating text output
- > Can accidentally reveal sensitive training data
- > ...?





# Thoughts on using AI for forensic purposes

- > Hansken is:
  - used for investigations, bit
  - designed for evidentiary use
- > Evidentiary use is more strict:
  - Accurate
  - Repeatable
  - Reproducible
- > So algorithms must be:
  - Explainable
  - Validated
  - Deterministic
  - Not depend on external data
- > Artificial Intelligence:
  - Use external data: Training sets
  - Use external data: Cause bias
  - Lacks explainability
- > Use AI for investigative purposes, with
  - Disclaimer
  - Education
- > By the way:
  - Not all currently used algorithms are good
  - Data under investigation can results from AI itself





# Hallucinations, data privacy and explainability

## > Preventing hallucinations:

- Provide clear prompts to ChatGPT to base its response on digital traces
- ChatGPT should not hallucinate but inform that there are no relevant traces
- Retrieval-Augmented-Generation (RAG) comes to the rescue
- Explicit references to the source on which a response is based

## > Maintaining data privacy:

- Digital traces and case specific details can not be send to the public cloud (e.g., ChatGPT in the OpenAI cloud)
- Powerfull Large Language Models can already be deployed on premise (e.g., Meta's Llama 2)
- Assumption: Open source LLMs with RAG do not need the extensive factual knowledge as ChatGPT/GPT-4

## > Explaining responses:

- Identify the sources that were retrieved as part of the RAG method to explain the response
- Reproducibility over creativity (experiment with "temperature" of the LLM)

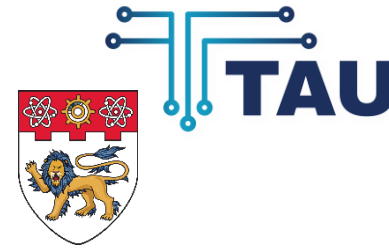


# Topics for future work

- › Can we do this off-line with the same quality?
- › Build a co-pilot in Hansken leveraging Retrieval Augmented Generation (RAG)
- › Evaluate with (real) users
- › Advanced topics:
  - Multi-modal generative transformers (Visual ChatGPT)
  - Augmented language models
  - Planning an investigation



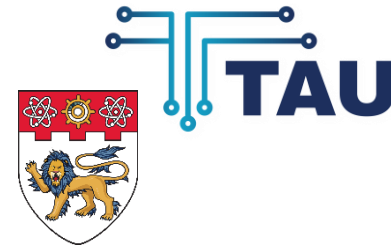
Midjourney prompt: Looking in a crystal ball seeing the future of artificial intelligence, ultra HD, super realistic, cinematic lighting. (fast)



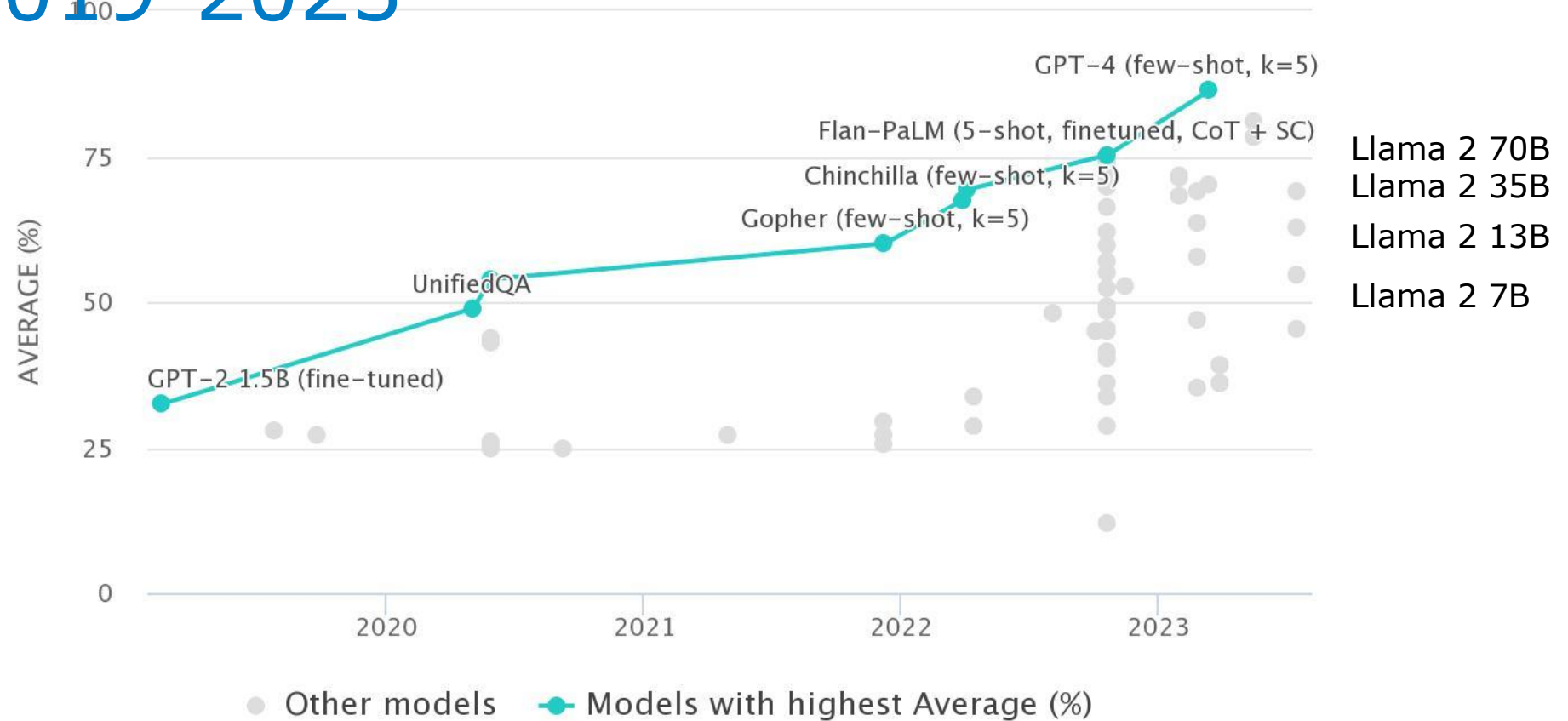
# How smart are LLMs?

- > Hugging Face **open** LLM leaderboard:
  - [https://huggingface.co/spaces/HuggingFaceH4/open\\_llm\\_leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard)
- > Score gebaseerd op:
  - **ARC**: Abstraction en Reasoning Challenge
  - **HellaSwag**: een benchmark die zich richt op gezond verstand redeneren
  - **MMLU**: Massive Multitask Language Understanding
  - **TruthfulQA**: een benchmark die beoordeelt of een taalmodel waarheidsgetrouwe antwoorden genereert

| Model                                  | Score |
|--|-------|
| garage-bAInd/Platypus2-70B-instruct    | 73.13 |
| upstage/Llama-2-70b-instruct-v2        | 72.95 |
| fangloveskari/Platypus_QLoRA_LLaMA_70b | 72.94 |
| yeontaek/llama-2-70B-ensemble-v5       | 72.86 |
| TheBloke/Genz-70b-GPTQ                 | 72.82 |
| TheBloke/Platypus2-70B-Instruct-GPTQ   | 72.81 |
| psmathur/model_007                     | 72.72 |
| yeontaek/llama-2-70B-ensemble-v4       | 72.64 |
| psmathur/orca_mini_v3_70b              | 72.64 |
| ehartford/Samantha-1.11-70b            | 72.61 |
| MayaPH/GodziLLa2-70B                   | 72.59 |
| psmathur/model_007_v2                  | 72.49 |
| chargoddard/MelangeA-70b               | 72.43 |
| ehartford/Samantha-1.1-70b             | 72.42 |
| psmathur/model_009                     | 72.36 |



# MMLU 2019-2023



<https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu>

# How open are LLMs?

- › Researchers from Nijmegen University:
  - Opening up ChatGPT: tracking "open source" LLM + RL
  - <https://opening-up-chatgpt.github.io/>

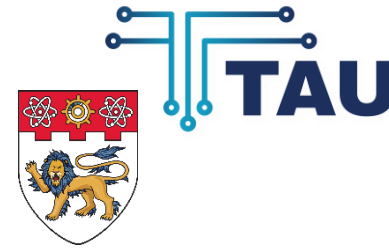
There is a growing amount of instruction-tuned text generators billing themselves as 'open source'. How open are they really? [paper](#) [logo](#)

| Project (maker, bases, URL) | Availability |          |             |           | Documentation |         |      |              |          |       | Access    |         |         |     |
|-----------------------------|--------------|----------|-------------|-----------|---------------|---------|------|--------------|----------|-------|-----------|---------|---------|-----|
|                             | Open code    | LLM data | LLM weights | RLHF data | RLHF weights  | License | Code | Architecture | Preprint | Paper | Modelcard | Dataset | Package | API |
| BLOOMZ                      | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| sigprince-workshop          | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| Pythia-Chat Base-7...       | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| legitthecomputer            | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| Open Assistant              | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| LAION-AI                    | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| dolly                       | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| anyscale                    | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| RedPajama-INCITE            | ~            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ~   |
| logothic/legit              | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| litix                       | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| logothic                    | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| MPT-7B Instruct             | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| MosaicML                    | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| MPT-30B Instruct            | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| MosaicML                    | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| Vicuna 13B v1.3             | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| LMSYS                       | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| minChatGPT                  | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| nlphero/gpt                 | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| ChatRWKV                    | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| 00000000000                 | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| OpenChat V3                 | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| OpenChat                    | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| Centras-GPT-11M             | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |
| Centras 1 - Schermer        | ✓            | ✓        | ✓           | ✓         | ✓             | ~       | ~    | ✓            | ✓        | ✓     | ✓         | ✓       | ✓       | ✓   |

|                          |                     |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--------------------------|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CarperAI                 | LLM base: LLaMA     | RL base: OASST1 (human), GPT4All (h...  | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
| Stanford Alpaca          | ✓                   | X                                       | ~ | ~ | ~ | X | ~ | ✓ | X | X | X | X | X | ~ |
| Stanford University CRFM | LLM base: LLaMA     | RL base: Self-Instruct (synthetic)      | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
| Koala 13B                | ✓                   | ~                                       | ~ | ~ | X | ~ | ~ | ~ | X | X | X | X | X | ~ |
| BAIR                     | LLM base: LLaMA 13B | RL base: HC3, ShareGPT, alpaca (synt... | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
| LLaMA2 Chat              | X                   | X                                       | ~ | X | ~ | X | X | ~ | ~ | X | ~ | X | X | ~ |
| Facebook Research        | LLM base: LLaMA2    | RL base: Meta, StackExchange, Anthropic | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
| ChatGPT                  | X                   | X                                       | X | X | X | X | X | X | ~ | X | ~ | X | X | X |
| OpenAI                   | LLM base: GPT 3.5   | RL base: Instruct-GPT                   | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |

How to use this table: Every cell records a three-level openness judgement (✓ open, ~ partial or X closed) with a direct link to the available evidence; on hover, the cell will display the notes we have on file for that judgement. At the end of a row, the § is a direct link to source data. The table is sorted by cumulative openness, where ✓ is 1, ~ is 0.5 and X is 0 points.

|            |                             |                               |   |   |   |   |   |   |   |   |   |   |   |   |
|------------|-----------------------------|-------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|
| databricks | LLM base: FleutherAI nvthia | RL base: databricks-dolly-15k | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
|------------|-----------------------------|-------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|



# Agenda

| Time  | Title  |
|-------|--|
| 13:00 | Part I: Introduction Large Language Models                 |
| 13:45 | Part II: Hands-on prompt engineering for digital forensics |
| 14:30 | Part III: Hands-on with a LLM in a Google Colab notebook   |
| 15:00 | Break  |
| 15:30 | Part III continued   |
| 16:00 | Part IV: Hands-on with Retrieval Augmented Generation      |
| 16:30 | Panel discussion on LLMs in the legal domain               |
| 17:00 | End  |





Netherlands Forensic Institute  
*Ministry of Justice and Security*



# Hands-on prompt engineering for digital forensics

Part II: Hans Henseler



# Github & Google CoLab

Link:

- › <https://github.com/HansHenseler/DFRWS-APAC-LLM-Workshop>

Notebooks:

- › Part II: Prompt engineering with ChatGPT for Digital Forensics
- › Part III: Handson with Llama2
- › Part IV: Retrieval Augmented Generation with Llama2

Requirements:

- › Google CoLab is free but you need a Gmail account!
- › Account for accessing free version of OpenAI ChatGPT
- › Make sure to select a T4 GPU



# Reductive operations

- > **Summarization** — Say the same thing with fewer words. Can use list, notes, executive summary.
- > **Distillation** — Purify the underlying principles or facts. Remove all the noise, extract axioms, foundations, etc.
- > **Extraction** — Retrieve specific kinds of information. Question answering, listing names, extracting dates, etc.
- > **Characterizing** — Describe the content of the text. Describe either the text as a whole, or within the subject.
- > **Analyzing** — Find patterns or evaluate against a framework. Structural analysis, rhetorical analysis, etc
- > **Evaluation** — Measuring, grading, or judging the content. Grading papers, evaluating against morals
- > **Critiquing** — Provide feedback within the context of the text. Provide recommendations for improvement



# Transformative Operations

- > **Reformatting** — Change the presentation only. Prose to screenplay, XML to JSON.
- > **Refactoring** — Achieve same results with more efficiency. Say the same exact thing, but differently.
- > **Language Change** — Translate between languages. English to Russian, C++ to Python.
- > **Restructuring** — Optimize structure for logical flow, etc. Change order, add or remove structure.
- > **Modification** — Rewrite copy to achieve different intention. Change tone, formality, diplomacy, style, etc.
- > **Clarification** — Make something more comprehensible. Embellish or more clearly articulate.



# Generative (Expansion) Operations

- > **Drafting** — Generate a draft of some kind of document. Code, fiction, legal copy, KB, science, storytelling.
- > **Planning** — Given parameters, come up with plans. Actions, projects, objectives, missions, constraints, context.
- > **Brainstorming** — Use imagine to list out possibilities. Ideation, exploration of possibilities, problem solving, hypothesizing.
- > **Amplification** — Articulate and explicate something further. Expanding and expounding, riffing on stuff.



# Prompt engineering with ChatGPT for DF

## Our 4 case study experiments:

1. Writing search queries
2. Summarising chat conversations
3. Analysing search results
4. Reverse engineering

Part II Colab we will focusses on #1, #3 and #4



Midjourney prompt: photorealistic picture of a digital sleuth in the style of Sherlock Holmes as a robot investigating a crime scene with digital traces in smartphones and computers (fast)



# Github & OpenAI ChatGPT

## Link:

- › <https://github.com/HansHenseler/DFRWS-APAC-LLM-Workshop>

## Notebooks:

- › Part II: Prompt engineering with ChatGPT for Digital Forensics
- › Part III: Handson with Llama2
- › Part IV: Retrieval Augmented Generation with Llama2

## Requirements:

- › You need to have an account to chat with ChatGPT 3.5 (free)
- › You can open the notebook in Google CoLab for better navigation



# More on prompt engineering

- > Videos and articles by David Shapiro:
  - <https://medium.com/@dave-shap/become-a-gpt-prompt-maestro-943986a93b81>
  - On YouTube: <https://www.youtube.com/watch?v=aq7fnqzeaPc>
  - About System Prompts:  
<https://www.youtube.com/watch?v=oILYjtbmLgc&t=760s>
- > Video and notebook by AssemblyAI:
  - Prompt Engineering 101
    - <https://www.youtube.com/watch?v=aOm75o2Z5-o>
  - Prompt\_Engineering\_101.ipynb
    - <https://colab.research.google.com/drive/1IHd9b8C4ccAGpkK06dzcFB0asjXWGZi0>





Netherlands Forensic Institute  
*Ministry of Justice and Security*



# Hands-on with a local LLM in a Google Colab notebook

Part III: Victor C.W. Cheng and Hans Henseler

## Part III: Hands-on with a LLM in a Google Colab notebook

---

October 2023



# How to get LLMs



## **Subscribe to OpenAI GPT4, Google PaLM**

- Models are generally more powerful (Higher scores in various assessments)
- No need to setup and maintain the models and hardware
- Need to pay
- Privacy problems

## **Setup a local in-house LLM**

- Many models are free
- No privacy issue
- Mid range hardware required
- Self maintenance (very limited support from publishers)

# Local LLMs



What  
**Hardware**  
is required?

What  
**Models**  
to be used?



# Background Info for Model Selection



# Background Info for Model Selection



## What do LLMs perform?

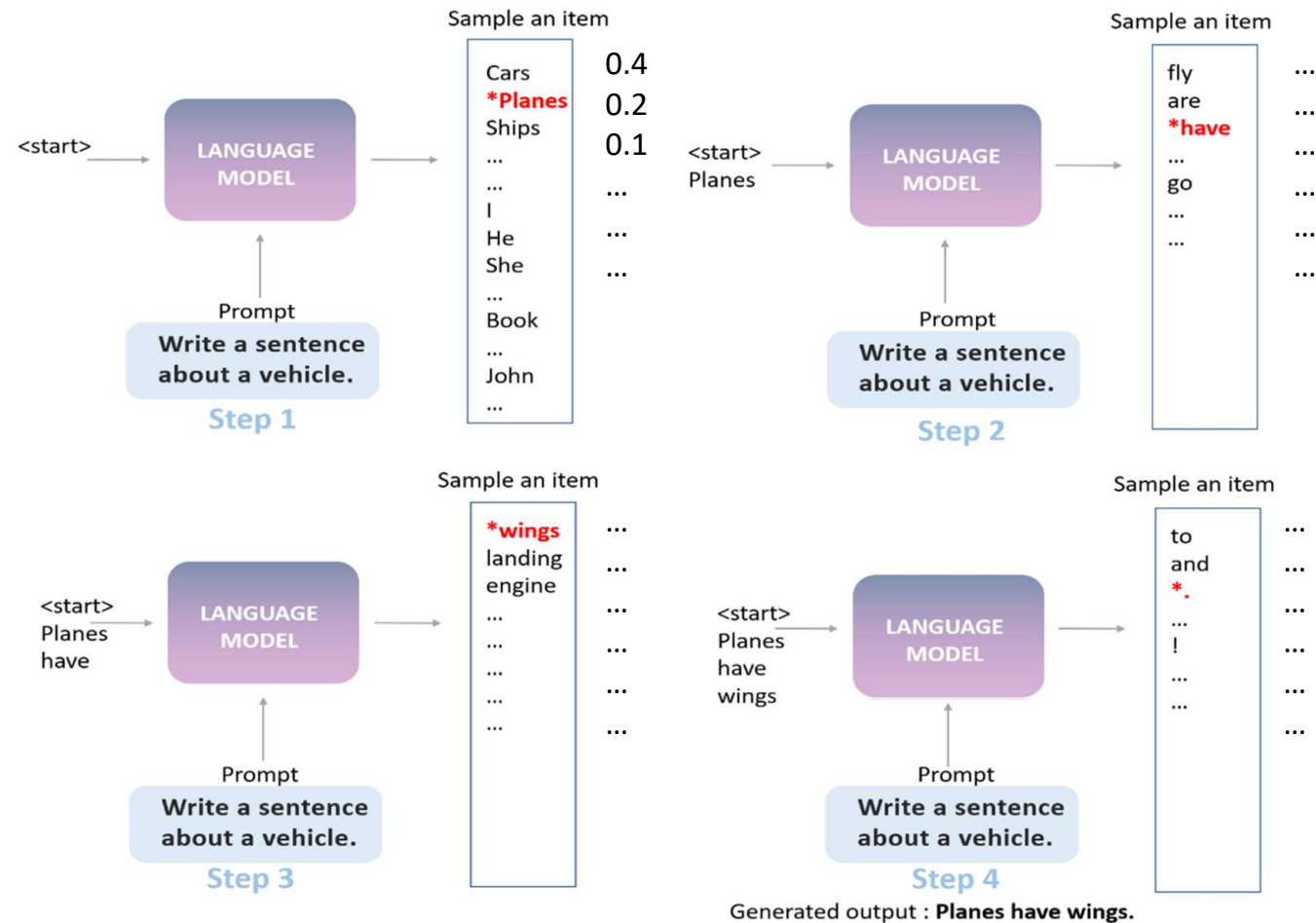
Generate coherent text (semantically related text) which can delivery meaningful contents.

$$P(w_n | w_{n-1}, w_{n-2}, \dots, \text{prompt})$$

Words are **probabilistically generated** one by one: depending on the previous generated words and the user given “prompt”. Different LLMs have different probability distribution functions!!!



# An example illustrating how LLMs generate words using the prompt: “*Write a sentence about a vehicle.*”



# Important Parameters for Local LLMs



|                          |  |
|--------------------------|--|
| <b>Top_k</b>             | Only consider the top $k$ words  |
| <b>Top_p</b>             | Only consider the top words having total probabilities $\leq Top_p$  |
| <b>Temperature</b>       | Higher value $\rightarrow$ more diverse and creative content, but content may not be coherent or even irrelevant |
| <b>n_ctx, max_length</b> | Max. context length  |
| <b>Max_new_tokens</b>    | Max. number of tokens to be generated  |
| <b>Repeat_penalty</b>    | Discourage repetitive or redundant output  |



# Local LLM Selection



## Features of local LLMs to be considered:

|  |   |
|--|---|
| <b>Size of the models<br/>(num. of<br/>parameters/weights)</b> | <ul style="list-style-type: none"><li>• 7B, 13B, 30B, etc.</li><li>• Larger size models usually give better performance but require better hardware and slower</li></ul>    |
| <b>Nature of the models</b>                                    | Use instruct model or chat models for Q&A and Retrieval Augmented Generation (RAG)  |
| <b>Weight Quantization</b>                                     | Usually map floating point values (16bits/32bits) to integer values (int8, int4, etc)   |
| <b>Model Data<br/>Format/Structure</b>                         | Hugging Face, GGUF, GGML (now replaced by GGUF), GPTQ, AWQ  |
| <b>Context length<br/>(tokens)</b>                             | <ul style="list-style-type: none"><li>• 2K, 4K, 8K, 32K....</li><li>• ChatGPT : 8K, GPT4: 32K</li><li>• Number of context words ~ (0.6 or 0.7) * number of tokens</li></ul> |

# Parameter Quantization



Models are **too big!**

High VRAM GPU cards are **too expensive!** Almost no competitor !!!

**Limited supply** of high VRAM GPUs

Model computation is **slow!**

How to make models smaller, while preserving the number of parameters/weights, or minimizing the degradation of performance?



Use **smaller number of bits** to store the parameters/weights

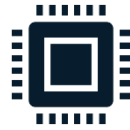
Float32, float16 ---→ int8 (8-bit integer), int4, ...

**Faster** computation

# Frameworks



**Hugging Face:**  
Traditional  
framework



**GGUF/GGML:**  
Optimized for  
CPU and (CPU +  
GPU)



**GPTQ:**  
Optimized for  
GPU and (GPU +  
CPU)



**AWQ:**  
Recent efficient  
quantization method  
(size, speed)

Publisher/model-name model-size model-type framework context-size

[meta-llama/Llama-2-7b-chat-hf](#)

Text Generation • Updated Aug 9 • ↓ 1.09M • ♥ 1.32k

[meta-llama/Llama-2-7b](#)

Text Generation • Updated Jul 20 • ♥ 2.65k

[meta-llama/Llama-2-70b-chat-hf](#)

Text Generation • Updated Aug 9 • ↓ 141k • ♥ 1.39k

[meta-llama/Llama-2-7b-hf](#)

Text Generation • Updated Aug 9 • ↓ 563k • ♥ 627

[meta-llama/Llama-2-13b-chat-hf](#)

Text Generation • Updated Aug 9 • ↓ 240k • ♥ 585

[TheBloke/Llama-2-7B-Chat-GGML](#)

Text Generation • Updated 6 days ago • ↓ 7.04k • ♥ 570

## For example:

Llama 2 7B chat model

No. of parameters: 7B (float 16)

Memory required: ~ 14GB

Q4\_0: 4bit quantization

7B parameters ~ 3.5GB

Q5\_0: 5bit quantization


7B parameters ~ 4.4GB









































Q6\_K\_S: 6bit K-quantization

7B parameters ~ 5.3GB

Q8\_0: 8bit quantization

7B parameters ~ 7.0GB



|   |         |   |   |
|---|---------|---|---|
|  llama-2-7b-chat.ggmlv3.q4_0.bin      | 3.79 GB |    |    |
|  llama-2-7b-chat.ggmlv3.q4_1.bin      | 4.21 GB |    |    |
|  llama-2-7b-chat.ggmlv3.q4_K_M.bin    | 4.08 GB |    |    |
|  llama-2-7b-chat.ggmlv3.q4_K_S.bin    | 3.83 GB |    |    |
|  llama-2-7b-chat.ggmlv3.q5_0.bin      | 4.63 GB |    |    |
|  llama-2-7b-chat.ggmlv3.q5_1.bin      | 5.06 GB |    |    |
|  llama-2-7b-chat.ggmlv3.q5_K_M.bin    | 4.78 GB |    |    |
|  llama-2-7b-chat.ggmlv3.q5_K_S.bin    | 4.65 GB |   |   |
|  llama-2-7b-chat.ggmlv3.q6_K.bin  | 5.53 GB |  |  |
|  llama-2-7b-chat.ggmlv3.q8_0.bin  | 7.16 GB |  |  |



# Github & Google CoLab

Link:

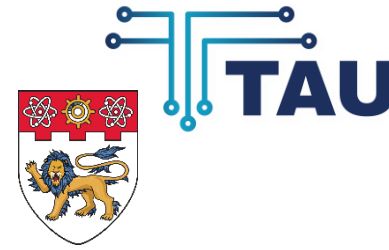
- › <https://github.com/HansHenseler/DFRWS-APAC-LLM-Workshop>

Notebooks:

- › Part II: Prompt engineering with ChatGPT for Digital Forensics
- › Part III: Hands on with Llama2
- › Part IV: Retrieval Augmented Generation with Llama2

Requirements:

- › Google CoLab is free but you need a Gmail account!
- › Make sure to select a T4 GPU



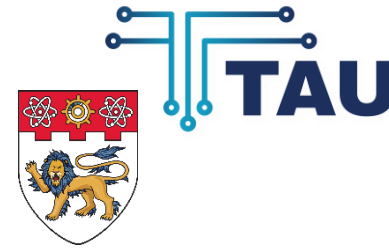
# Agenda

| Time  | Title  |
|-------|--|
| 13:00 | Part I: Introduction Large Language Models                 |
| 13:45 | Part II: Hands-on prompt engineering for digital forensics |
| 14:30 | Part III: Hands-on with a LLM in a Google Colab notebook   |
| 15:00 | Break  |
| 15:30 | Part III continued   |
| 16:00 | Part IV: Hands-on with Retrieval Augmented Generation      |
| 16:30 | Panel discussion on LLMs in the legal domain               |
| 17:00 | End  |



# Hands-on with a LLM in a Google Colab notebook

Part III continued



# Agenda

| Time  | Title  |
|-------|--|
| 13:00 | Part I: Introduction Large Language Models                 |
| 13:45 | Part II: Hands-on prompt engineering for digital forensics |
| 14:30 | Part III: Hands-on with a LLM in a Google Colab notebook   |
| 15:00 | Break  |
| 15:30 | Part III continued   |
| 16:00 | Part IV: Hands-on with Retrieval Augmented Generation      |
| 16:30 | Panel discussion on LLMs in the legal domain               |
| 17:00 | End  |





Netherlands Forensic Institute  
*Ministry of Justice and Security*

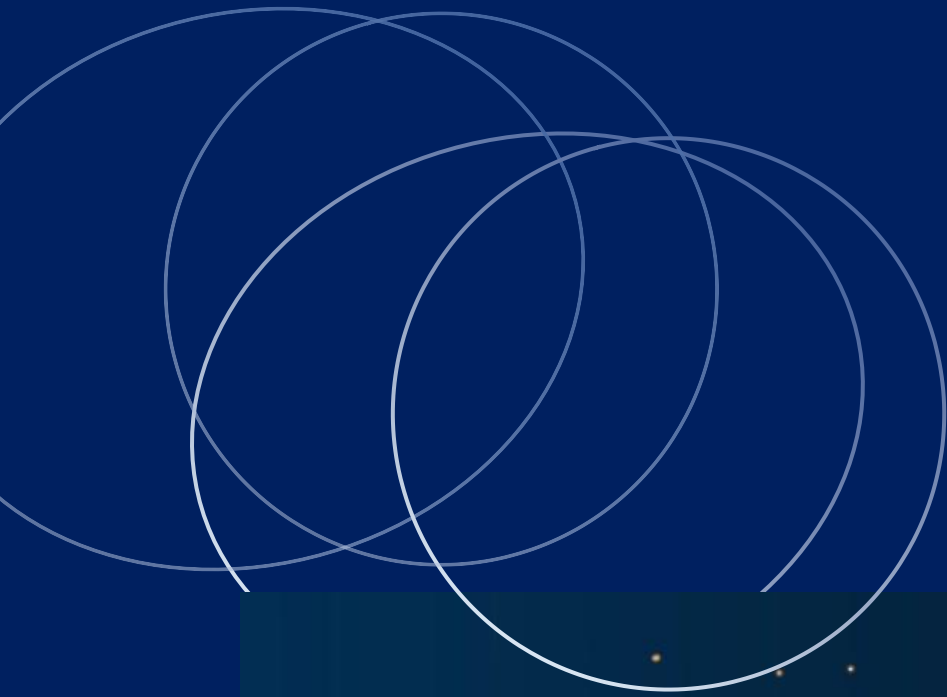


# Hands-on with Retrieval Augmented Generation

Part IV: Victor C.W. Cheng and Hans Henseler



## Part IV: Hands-on with Retrieval Augmented Generation (RAG)





## No GPU

- **GGML/GGUF models** are a good choice if no good GPU, but good CPU & 16GB RAM required
- optimized for CPU execution

## Small GPU card (e.g. 4GB or 6GB VRAM)

- off load part of the model weights/layers to GPU
- highly increase the word generation speed, compared to no off loading.

## Larger GPU card (e.g. 16GB VRAM)

- can consider GPTQ models which are optimized for GPU execution and usually have slightly smaller sizes compared with GGML/GGUF models.

## Huge GPU (e.g. V100 36GB, A100 40GB/80GB)

1. Using original models (no quantization) say 13B (26GB) or 30B (60GB) models, or
2. Larger size models\* (with quantization) say 70B models (36GB with 4bit quantization).

\* Option 2 usually gives better performance.

# Background



- LLMs are pretrained with public domain information.
- Public information may not be up-to-date and less accurate.
- Embedding new/unseen information after pretraining is difficult.
- “Hallucination” may happen and hard to detect
- Resolution:
  1. Fine tuning with new information or new tasks
  2. Embed the new information to the LLM input prompts (known as context) and instruct LLMs to respond based on the context, known as retrieval augmented generation (RAG).
- Can be regarded as open book Q&A

# Background



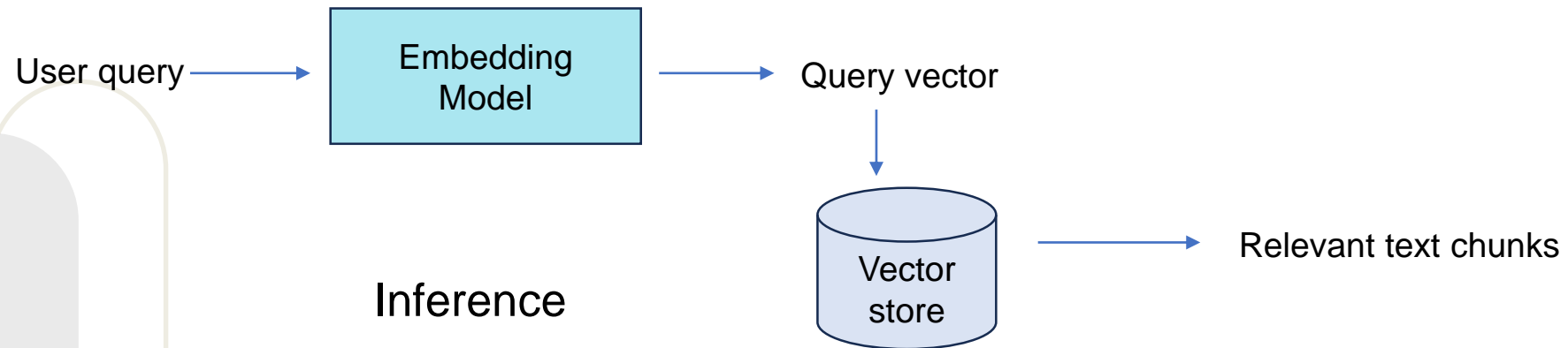
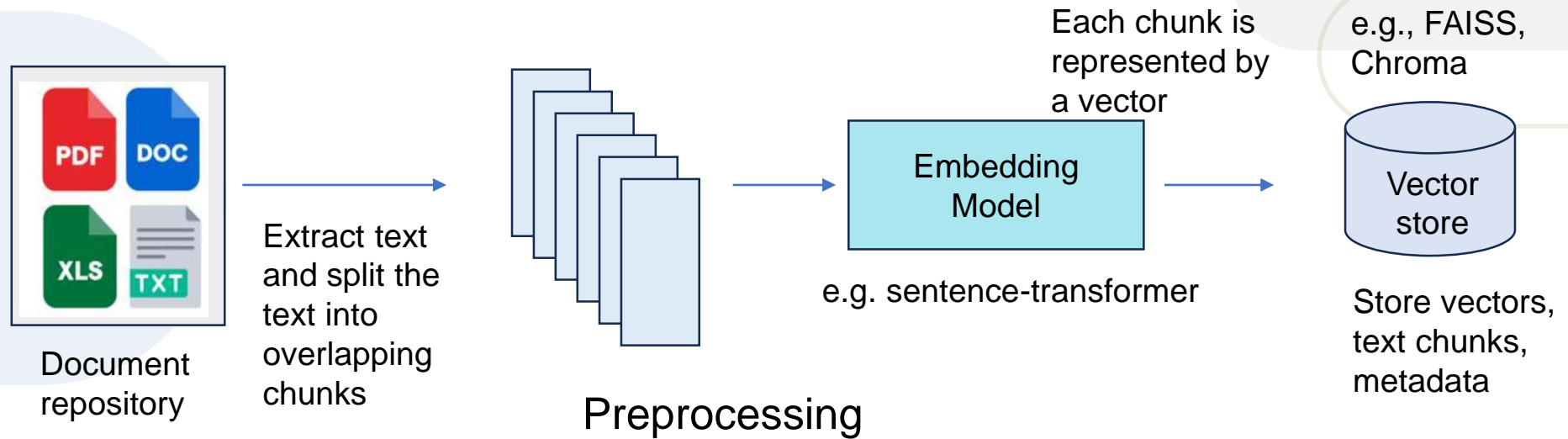
- RAG enable up-to-date and private information to be analyzed or processed by LLMs
- Hallucination is less probable (still be possible)
- Source of information can be identified and hence LLM's responses can be checked (but still need manual efforts), hence verifiable.
- No training or fine-tuning, thus low cost and time saving.
- RAG has 2 stages:
  1. Retrieval stage: search for relevant information
  2. Generation stage: Generate answers to the user questions

# Large information sources

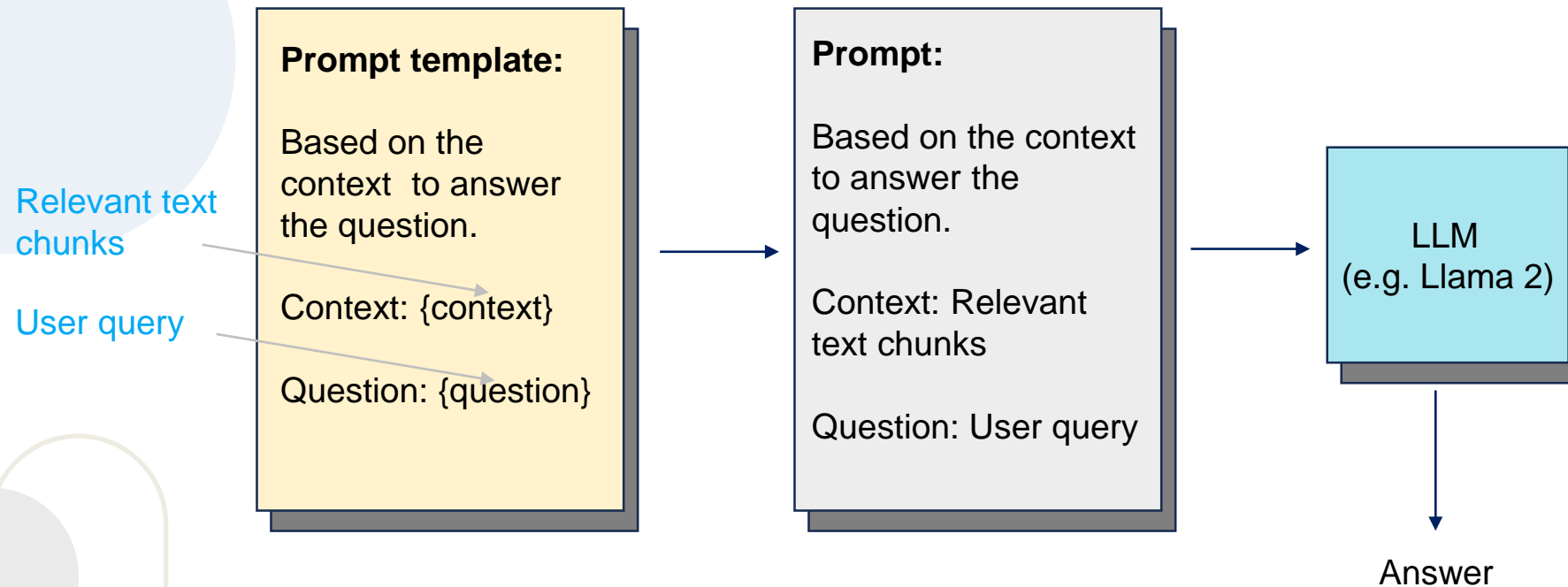


- Cannot pass huge size passages/documents to LLMs, due to limited context length they can handle, e.g. GPT4: 32K tokens, Llama 2: 4K tokens.
- Most LLMs only accept text format information.
- If too many documents or the document is too large, split them into individual overlapping chunks of text.
- Hence,
  - Step 1: Identify relevant chunks of text.
  - Step 2: Create a prompt to include the relevant chunks together with the user query and send to the LLM.

# Identify Relevant Chunks



# Create prompt and get responses





# Challenges



- Selection of correct chunks is extremely important → incorrect chunks will result in inaccurate answers or no answer at all.
- If queries involve information spanning several chunks, selecting the correct chunks becomes very difficult.
- Tables, especially those extracted from PDFs, can be challenging for LLMs to comprehend.
- Most local LLMs currently lack support for images.



# Github & Google CoLab

## Link:

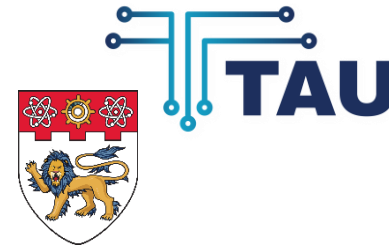
- › <https://github.com/HansHenseler/DFRWS-APAC-LLM-Workshop>

## Notebooks:

- › Part II: Prompt engineering with ChatGPT for Digital Forensics
- › Part III: Handson with Llama2
- › Part IV: Retrieval Augmented Generation with Llama2

## Requirements:

- › Google CoLab is free but you need a Gmail account!
- › Make sure to select a T4 GPU



# Agenda

| Time  | Title  |
|-------|--|
| 13:00 | Part I: Introduction Large Language Models                 |
| 13:45 | Part II: Hands-on prompt engineering for digital forensics |
| 14:30 | Part III: Hands-on with a LLM in a Google Colab notebook   |
| 15:00 | Break  |
| 15:30 | Part III continued   |
| 16:00 | Part IV: Hands-on with Retrieval Augmented Generation      |
| 16:30 | Panel discussion on LLMs in the legal domain               |
| 17:00 | End  |



Netherlands Forensic Institute  
*Ministry of Justice and Security*



# Panel Discussion on LLMs in digital forensics and legal applications

Panel: Kwok Yan Lam and Victor C.W. Cheng

Moderator: Hans Henseler

# Thank you!



Netherlands Forensic Institute  
Ministry of Justice and Security



Hans Henseler  
h.henseler@nfi.nl

Kwok-Yan Lam  
kwokyan.lam@ntu.edu.sg

Zee Kin Yeong  
Yeong\_zee\_kin@sal.org.sg

Victor C.W. Cheng  
victor.cheng@tauexpress.com

## Published papers and articles:

**Applying Large Language Models for Enhancing Contract Drafting**  
Kwok-Yan Lam<sup>1</sup>, Victor C.W. Cheng<sup>2</sup> and Zee Kin Yeong<sup>3</sup>

<sup>1</sup> Nanyang Technological University, Singapore  
<sup>2</sup> TAU Express Pte Ltd, Singapore  
<sup>3</sup> Singapore Academy of Law, Singapore


**Abstract**  
This paper investigates the use of traditional AI and generative AI techniques in enhancing the work of legal professionals. We propose an approach that applies a combination of AI techniques, traditional AI augmented with generative models for automating some of the laborious tasks in contract drafting. With the launches of advanced AI models such as ChatGPT, legal professionals are anticipating how such technologies can streamline their works. We first introduce how these models generate text contents given a user prompt. Then we propose some practical approaches in "prompt writing" which enable better and more coherent contract clauses to be generated. As privacy is typically a great concern in using ChatGPT in professional domains, we also explore the feasibility and effectiveness of using on-premises Large Language Models (LLMs) such as "Vicuna" as practical alternatives that may address the privacy issues while producing acceptable performance in contract drafting. Since AI generated clauses may not match the strict legal requirements or even be incorrect, we propose an approach to evaluate the clauses with traditional AI by using sentence transformers to retrieve similar clauses from a trusted source and perform automatic content similarity analysis. Experimental results using the public dataset LEDGAR showed that LLMs are useful tools for contract clause drafting and the automated comparison results can work as hints or recommendations that users can consider to revise and enhance the generated clauses, hence simplifying the task of contract drafting by legal professionals in an augmented intelligence manner.

**Keywords**  
LLMs, ChatGPT, clause recommender, contract drafting, hallucination, AI safety

**1. Introduction**  
Contracts are legally binding between parties and accurately capture the agreement between the parties and thus pose a great challenge to researchers. In general, contract drafts typically

<https://ceur-ws.org/Vol-3423>

**ChatGPT: A Digital Sleuth For Detectives?**  
21st February 2023 by Forensic Focus




By Hans Henseler, Professor of Digital Forensics & E-Discovery, University of Leiden Applied Sciences, and Senior Digital Forensic Scientist at the Netherlands Forensic Institute.

Helping to formulate search questions

<https://www.forensicfocus.com/articles/chatgpt-a-digital-sleuth-for-detectives>

eForensics  
HOME / NEW EDITION  
UNRAVELING DIGITAL MYSTERIES: HOW AI COPILOTS CAN REVOLUTIONIZE DIGITAL FORENSIC INVESTIGATIONS

**Unraveling Digital Mysteries: How AI Copilots can Revolutionize Digital Forensic Investigations**



By Hans Henseler, Professor of Digital Forensics & E-Discovery, University of Leiden Applied Sciences, and Senior Digital Forensic Scientist at the Netherlands Forensic Institute. Introduction In hindsight, 2021 was a significant inflection point in the world of artificial intelligence, characterized by remarkable developments in deep learning, manifesting in models such as...

<https://eforensicsmag.com/unraveling-digital-mysteries-how-ai-copilots-can-revolutionize-digital-forensic-investigations/>

**ChatGPT as a Copilot for Investigating Digital Evidence**  
Hans Henseler<sup>1,2</sup>, Harm van Beek<sup>2</sup>

<sup>1</sup>University of Applied Sciences Leiden, The Netherlands  
<sup>2</sup>Netherlands Forensic Institute

**Abstract**  
In today's technology-driven legal landscape, practitioners must continually adapt to new tools and methods that aid not only in addressing cybercrime but also in managing traditional crimes with digital components. This paper explores the potential of advanced AI-powered solutions, such as ChatGPT, in enhancing the capabilities of investigators in various aspects of their investigations. We delve into three specific applications pertinent to legal professionals: (1) writing structured queries utilizing natural language and trace models, (2) summarizing, evaluating, and visualizing electronic communications, and (3) analyzing search results. Our findings demonstrate that once ChatGPT is proficient in the query language and data model of the system containing the digital evidence, it holds significant promise in assisting legal professionals in conducting effective investigations.

**Keywords**  
digital forensics, eDiscovery, large language models, natural language processing, deep learning, chatgpt, gpt-4

**1. Introduction**  
The legal profession is witnessing a significant surge in the adoption of artificial intelligence (AI) tools, with ChatGPT emerging as a prominent development since November 2022 [1]. Powered by OpenAI's advanced large language model, ChatGPT offers a natural and engaging conversational interface on an extensive array of topics encountered during its training. ChatGPT's web application provides users with access to various models, including the Default GPT-3.5 turbo (a refined and superior version of GPT-3), Legacy GPT-3.5 (the preceding ChatGPT model), and GPT-4 (the most sophisticated model, exclusively accessible to ChatGPT Plus subscribers). The experiments discussed in this paper employ the ChatGPT/GPT-4 model, which showcases its potential applications in the domain of digital evidence investigation. ChatGPT has been fine-tuned with Reinforcement Learning from Human Feedback (RLHF) to ensure its responses are helpful, harmless, and honest. In this paper we describe the rise of ChatGPT followed

<https://ceur-ws.org/Vol-3423>