Filière Informatique de gestion
Studiengang Wirtschaftsinformatik

Hes·so VALAIS WALLIS
Haute Ecole de Gestion
Hochschule für Wirtschaft

Bachelor thesis 2023

Integration of Artificial intelligence for the exploration of energy transition solutions

Not confidential

Student : Abdullah / Binjos

Professor : David / Wannier

Filling date: 28 July 2023

Reference of the first image

https://hotpot.ai/art-generator, image created by an AI

Abstract & Foreword

This bachelor thesis closes my formation at the HES-SO Valais for the obtention of the bachelor in business IT, this work has been done during the last semester of my formation during my cursus of 3 years.

The subject of the bachelor thesis has been given by my professor Mr. Wannier David in collaboration with the EPFL of Sion with Mr. Lopez Michel and Prof. Maréchal François.

Keywords: Artificial intelligence, Deep learning, OpenAI, LLaMA, LoRA, PEFT, Embedding, Finetuning.

Thanks

I'd like to express my profound gratitude to Michel, who has been an invaluable source of assistance during my bachelor's thesis. His guidance and advice have shed light on the process of writing a strong report and provided direction when I found myself at an impasse.

My thanks also go out to Prof. François Maréchal, whose insightful feedback on the proof of concept helped enhance its quality significantly.

I would also like to acknowledge my professor, David Wannier, for his corrective inputs and expert advice on my bachelor's thesis.

Finally, I dedicate this report to my father, with the hope that he takes pride in the person his son has become.

Table des matières

List of tables and figures

Abreviations list

NLP = Natural language processing

GPT = Generative Pre-trained Transformer

LLM = Large Language Model

ML = Machine Learning

ANN = Artificial Neural Network

LoRA = Low-Rank Adaptation

PEFT = Parameter-Efficient Fine-Tuning

LLaMA = Large Language Model Meta AI

Introduction

In 2023, the question about energy has become an important subject, the countries are developing at a fast rate and the production of energy must match the demand.

With the help of the Industrial Process and Energy Systems Engineering (IPESE) tools created by the EPFL, we are able to predict and generate based on complex algorithms and the inputs from the user energies scenarios for a country.

The user can analyze then the results and compare it with others to identify the optimal configuration for his scenario.

Deliverables

For this bachelor thesis project, the project will be available on a ZIP for the product owner.

Problematic

With the arrival of the artificial intelligence (AI), the utilization of it is more and more common, the goal of this bachelor thesis is to research the different AI available on the market, compare them with theirs advantages and limitations and choose one to integrate it to the IPESE tool. (Galav, 2022)

Through the integration of AI into the IPESE tool, users can engage in an interactive dialogue with the AI. They can pose questions about various tool inputs to deepen their understanding, inquire about the outcomes, and seek ways to enhance them. The AI has the capability to reference its information sources and retain a memory of the interactions with the user.

By incorporating artificial intelligence into EnergyScope, the scenario creation process can potentially be simplified and more efficient

1 What is Artificial Intelligence

Artificial intelligence or AI is the simulation of human intelligence in machines that are programmed to do tasks that require human intelligence. With the advancement in technologies such as machine learning and deep learning in recent years, the AI systems can now learn from the precedents experiences and adapt to new information. (What Is Artificial Intelligence (AI)?, n.d. / What Is Artificial Intelligence and How Does AI Work?, Ed Burns.)

Theses AI are designed for:

- Analysis and interpretation of large data.
- Recognition of patterns
- Making decision based on the given data.

The usage of artificial intelligence is large, it impacts various industries such as healthcare, finance, transportation, education etc. (The Impact of Artificial Intelligence on 5 Industries, n.d.)

With the recent technological advancement, AI is experiencing a significant surge in popularity and growth that led to the emergence and development of various applications such as:

- Virtual assistant
- Recommendation systems
- Autonomous vehicles
- Image recognition
- Automatization of processes

1.1 Generative AI vs discriminative AI

To better understand the contents of the following pages, it would be nice to familiarize yourself with the technical terms to ensure you don't feel confused.

Generative AI and discriminative AI are two approaches in the AI field and each has its own objective.

1.1.1 Generative AI

Generative AI is more concerned in creating new data that has similarities to the training data. It involves creating new instances that follow the same patterns has the training data, generative AI can be used for tasks like (Yes, Machines Make Mistakes: The 10 Biggest Flaws in Generative AI, n.d. / What Is Generative AI?, n.d.):

- Image generation
- Text synthesis

1.1.2 Discriminative AI

On the other hand, discriminative AI is aimed to distinguish different instances based on their traits / characteristics.

Discriminative AI models learn the decision boundaries between classes or categories of data, they aim to accurately assign the inputs into their respective classes and therefore do predictions or classifications based on the training data. (GOYAL, 2021)

It is commonly used for tasks like:

- Image classification
- Speech recognition

1.2 Learning types

Different types of learning exist for training and improving the AI model, each one of them has its own advantages and serve a specific use case. Three common types of learning are used in the field of AI.

1.2.1 Supervised learning

Supervised learning is a machine learning approach where a model learns from labelled data in the training dataset. (What Is Supervised Learning?, n.d.)

The objective of the model is to learn the correlations between the input data and the target labels so it can make a prediction from it. Supervised learning techniques need external supervision to train the models compared to the two others. (Python Simplified, 2021)

The main goal of this learning is:

- Classification
  - Predict in which class is the given image.
    - Example: knowing if the given picture is of a cat or a dog
- Regression
  - Used for numerical prediction.
    - Example: prediction of salary based on multiples independent variables such as
      - Age
      - Number of years in the company
      - Educational background

Let's say for example we have a model used for image classification, based on supervised learning, the model will be trained using a dataset of images containing their corresponding labels indicating the category of each image. (Seldon, 2022)

1.2.2 Unsupervised learning

Unsupervised learning is a machine learning approach where a model learns patterns and structures from the data without explicit labels or target outputs. The model explores the structure and try to discover patterns and relationships among the input variables. (Johnson, 2023)

The main goal of this type of learning is:

- Clustering
  - Make groups of similar experiences / similarities
    - Example: Group clients who have the same interests
- Associations
  - Look for relationships between variables in the data.
    - Example: To know which items are often bought together
- Dimension reduction
  - Reduce the number of variables while preserving as much information as possible.
    - Example: improvement of picture quality (7. Unsupervised Learning, n.d.)

1.2.3 Reinforced learning

Reinforcement learning is an ML approach but differentiates itself from the others by being an agent, the agent learns to interact with the given environment and can take actions in it. Each action is then rewarded or reprimanded and forces the agent to optimize its behavior to avoid penalties and maximize rewards. (Bhatt, 2019)

The main goal of the reinforcement learning is:

- Game Playing
- Training autonomous robot
    - Self-driving cars
    - Learning-based robots such as the AI agent of Deepmind that was able to reduce energy spending cost by 40% by managing the data centers. (DeepMind AI Reduces Google Data Centre Cooling Bill by 40%, n.d.)

1.3 Models

Models refers to the mathematical structures used by machine learning algorithms to make predictions / classifications / decisions based on the given data. (Klingler, 2023) AI models are trained on datasets to learn patterns / relationships and rules to be able to achieve for example, predictions on new and unseen data. (AI Model, 2022)

Foundational or pre-trained models

Foundational model is a large artificial intelligence model that is pre-trained on a big dataset using learning techniques (supervised / unsupervised etc.). They serve as a basis / starting point for a wide range of downstream tasks and can be finetuned to perform specific tasks better. (What Are Foundation Models? | IBM Research Blog, n.d.)

These models played a large role in the development of AI because they have democratized the access to pre-trained models. Instead of starting from scratch, researchers and the community now have the opportunity to use those models and therefore drastically reduce the time and resources needed to develop a whole AI system. (“Foundation Model AI,” 2023)

For example, after the leak of the LLaMA foundational models, the community and researchers published more than 850 finetuned models on Huggingface.com (Mahmood,2022) based on the pre-trained model. (Vincent, 2023)

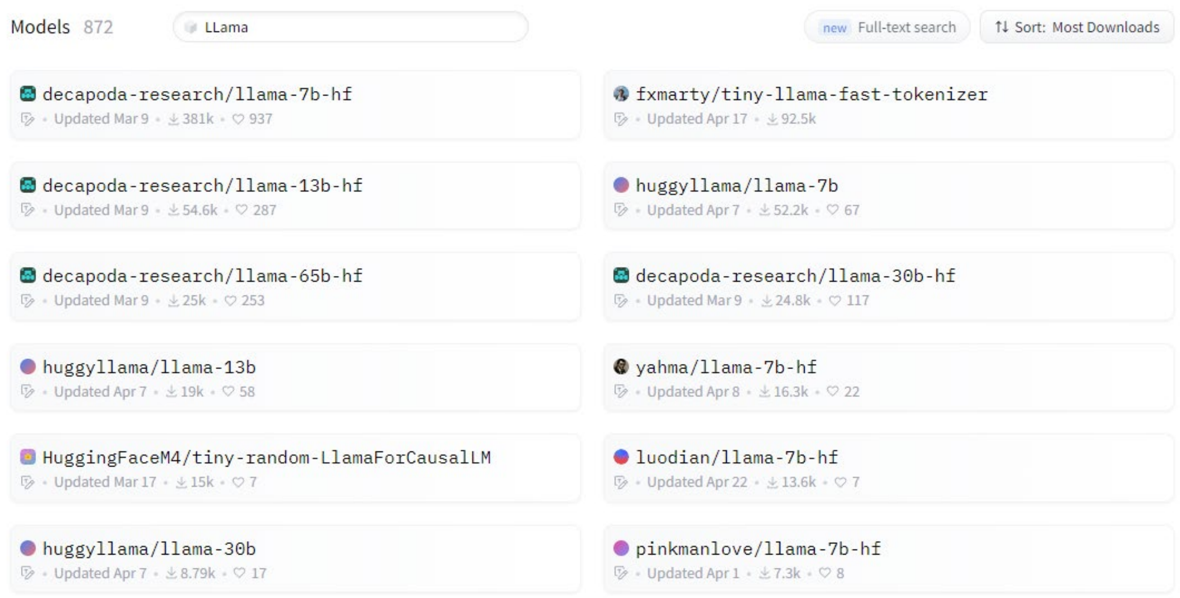


Figure 1 LLaMA models in Hugging face

Source: https://huggingface.co/

1.4 Weights

In a neural network, the weights refer to the parameters that determine the strength and influence of connection between neurons in a model. They are numerical values assigned to each connection between neurons in the network, showing the importance of one neuron to another. (Ganesh, 2022)

During the training process, the weights are adjusted iteratively to minimize the difference between the prediction and the desired result. The process of adjusting the weights is called “forward and back propagation” During the forward propagation, the training dataset is given trough the network, and the activations and outputs of each neuron are computed layers by layers. (Weight (Artificial

Neural Network), 2019)

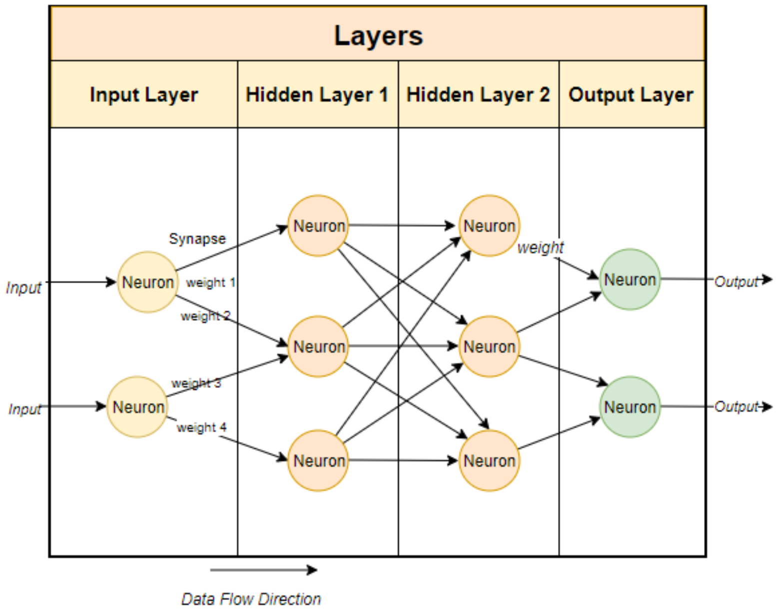


Figure 2 Layers in neural networks

Source: https://medium.com/fintechexplained/neural-networks-bias-and-weights-10b53e6285da

The process involves multiplying the input values by the corresponding weights, summing them up and applying the activation function, the result will then be passed to the next layer. Once the forward propagation is complete and a prediction has been made, using a loss function we calculate the difference between the prediction and the real value. (Bhargav, 2022)

Based on the optimizer algorithm, the loss is then propagated back through the network, layer by layer and adjust the weights of each connection. (Doshi, 2020)

1.5 Tokens

Tokens refer to the individual units or entities that make up a piece of text, It can be a word, character or even a sub-word. It depends on which tokenization method has been employed. (What Are AI Tokens?, n.d.)

Tokenization is taking the given sentence and breaking it up into individual words. (What Are Tokens and How to Count Them?, n.d.)

1.6 Common deficiencies

Despite the rapid evolution of LLM, there as several recurring deficiencies that can be observed when processing our requests. Despite the advancements, it is not uncommon to see shortcomings in the results provided by large language models.

1.6.1 Hallucinations

AI systems may sometimes show hallucinations, it means that the model generates text that goes beyond the given context, shows unrealistic patterns, and even completely forget the precedent context. (What Are AI Hallucinations and Why Are They a Problem?, n.d.)

Prompt example: “repeat after me “I love to eat hamburgers””
Hallucinated response:” I love to eat hamburgers while swimming in the saloon”

Figure 3 Prompt example

Source : author

There are several ways to counterpart those hallucinations by:

- limit the possible outcomes by avoiding open-ended questions.
- Force the AI to follow a template.
- Give the AI a specific role to follow.

1.6.2 Limited Generalization

Limited generalization occurs when the AI model cannot apply the learned knowledge on the new unseen data. It will perform well on the training data but fails to perform accurately on data that it hasn’t seen yet on this website. (Özpoyraz, 2022)

These deficiencies are often related to a phenomenon called “overfitting” when the model becomes too specialized to the specific patterns and making it therefore less flexible in handling new inputs. (Dezhic, 2017)

For example, the AI model can only recognize apples of a certain size and shape because he has only learned certain characteristics of an apple.

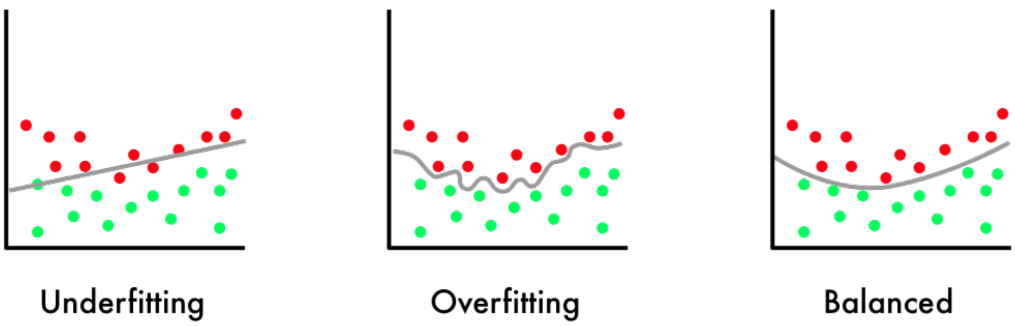


Figure 4 overfitting representation

Source: https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d

1.6.3 Data Limitation

Ai models rely a lot on large and diverse datasets for training. Depending on the field, it is sometimes difficult to gather high quality data and therefore limiting the performance of the AI model due to insufficient data availability. (Raizada, 2023)

Not pre-processing the dataset and filtering the bad information's is very counter-productive and must be a mandatory steps to avoid the AI models to learn knowledge that is biased. (AI Limitations in 2023, n.d.)

There are several ways to counterpart those data limitation by:

- Collect a substantial amount of data.
    - Must be diverse and comprehensive so the model can easily learn patterns from it

- Integrating data filtering before giving it to the AI model is essential to eliminate biased or erroneous information from the dataset.

1.6.4 Contextual understanding

AI systems are still frequently encountering difficulties in understanding the context given, signifying capturing the deeper meaning behind the text given. (Brdiczka, n.d.)

Difficulty from context understanding comes from several reasons:

- Ambiguity
    - Using words or sentences that have multiples meaning.
        - Example: "Time flies like an arrow"
    - Language and cultural nuance
        - Some words might have opposite meanings depending on the language used.
            - Example: "Gift" is often a positive connotation in most contexts but in certain cultures it can be interpreted as bribery.

- Background knowledge only know by the human.
    - Referring to knowledge gained through human experience that the AI does not have because it has been trained on a specific dataset.

As we will see in the following chapters, the context windows of the different AI models will exponentially grow release after release. (For example, GPT-4 (Terrasi, 2023) had increased by 40%

his context window) (Overcoming AI's Limitations to Reach True Understanding, 2022)

1.7 Quantization, the art of shrinking

Quantization is a fundamental technique in machine learning that aims to reduce a model size so it can run on smaller and less powerful devices. (Quantization, n.d.)

One of the approaches of quantization is to convert floating points values into integers, by reducing the precision we also decrease the memory usage while increasing the computation speed. (Nicholls, 2018)

While there is a diminution of precision and increase of information loss, the trade-off is almost always positive since the benefits from the reduction of memory and increase of computation speed outweigh the accuracy problem. (The Ultimate Guide to Deep Learning Model Quantization and Quantization-Aware Training, n.d.)

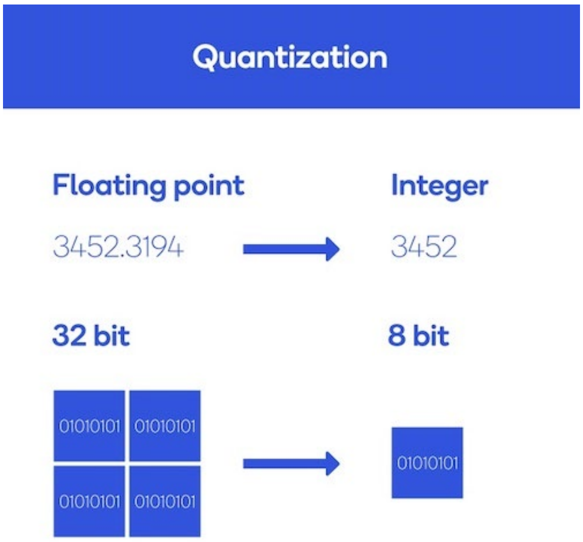


Figure 5 Quantization representation
Source: https://www.allaboutcircuits.com/technical-articles/neural-network-quantization-what-is-it-and-how-does-it-relate-to-tiny-machine-learning/#:~:text=Quantization%20shrinks%20neural%20networks%20by,latency%20and%20better%20power%20efficienc

4 bits / 8 bits

When you search for models on the internet, you will some of them with the inscription " 4bits" or "8bits". Let's say for example you want to download a model that has 7 billion parameters that has been launched in a 16 bit format. (A Gentle Introduction to 8-Bit Matrix Multiplication for Transformers at Scale Using Transformers, Accelerate and Bitsandbytes, n.d.)

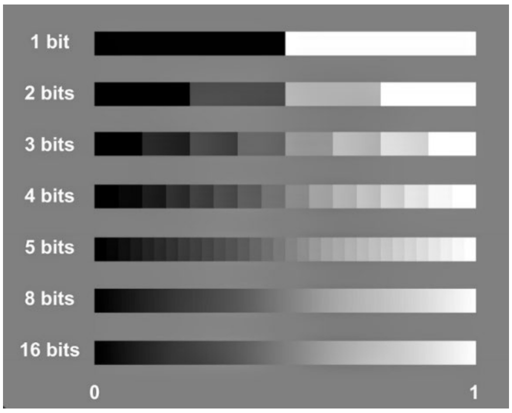


Figure 6 Bits comparison
Source : https://www.the-working-man.org/2014/12/bit-depth-color-precision-in-raster.html

It means that to run these 7 billion parameters model you will need 7 billion * 2 bytes = 14gb of ram.

We don't really need that level of precision and by shrinking the model we can achieve very good results, this is where the quantization become useful, by storing each weight from 16 bits into 8 bits, we significantly reduce the size and therefore the resources needed to use this model.

1.8 Finetuning

1.8.1 Definition

Fine-tuning is the process of taking a pre-trained model and applying more training on it by giving it a new dataset or tasks. It allows the model to adapt its learned knowledge and perform better in specific tasks. (Fine-Tuning a Neural Network Explained, n.d.)

1.8.2 When to use it?

Fine-tuning is typically employed for fundamental tasks like classification and conditional generation. The project's AI is expected to exhibit higher-order thinking, cite sources, and compare scenarios autonomously. These requirements are significantly complex for a model that has merely been fine-tuned.

1.8.3 OpenAI - Finetuning

Finetuning on OpenAI is available and the company provides a lot of documentation about it, it is not the same finetuning as we would normally do on a LLaMA model for example, but it is still effective. (OpenAI Platform, n.d.) The finetuning helps a model recognize new patterns and respond to them appropriately. (Khatik, 2023)

At the moment finetuning is available on theses following base models:

- Davinci
    - Can do any tasks other models can do
    - Higher quality of response
- Curie
    - Very capable
    - Faster
    - Low cost
- Baggage
    - Able to perfom simple task
    - Very fast
    - Low cost
- Ada
    - Able to perfom very simple tasks
    - Fastest model from all
    - Lowest cost from all

Finetuning an OpenAI model is effective in multiple scenarios and can solve a variety of problems better than the base model if trained correctly it is effective on:

- Classification
    - Examples:
        - Sentiment Analysis.
        - Classification of emails.

- Conditional generation
    - Examples:
        - Create ads from a given product and description.
        - Customer chatbot.

1.8.3.1 Installation of the workspace

The first step is the installation of the OpenAI command-line interface, you can do it in any IDE you choose, for the research I used pycharm professional. (OpenAI Platform, n.d.)

pip install --upgrade openai

After installing the libraries needed, you must then generate your API key on the platform.openai.com/account/api-keys and save the key address somewhere safe. (Liam Ottley, 2023)

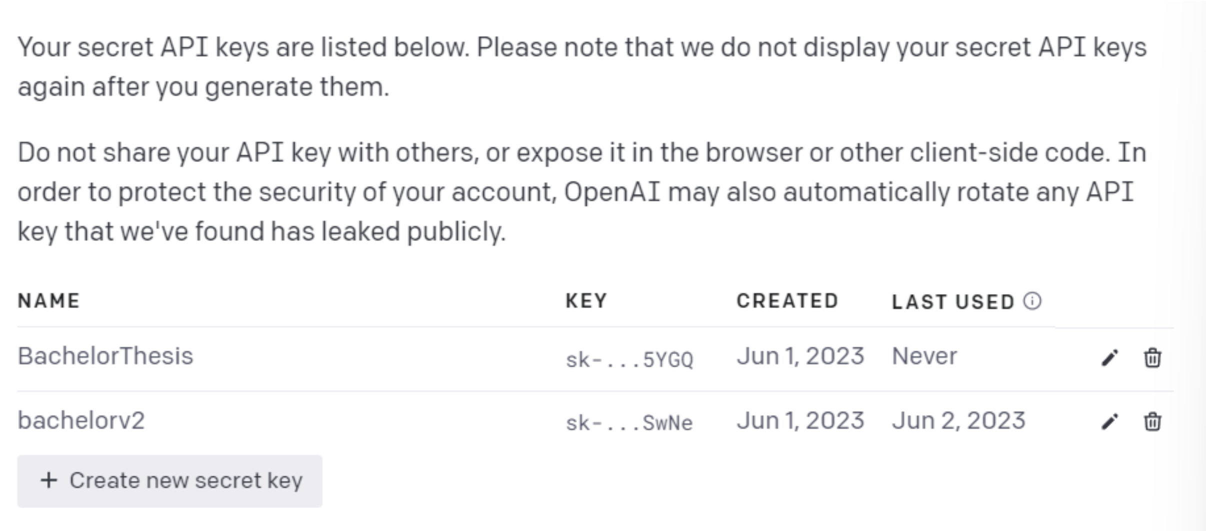


Figure 7 API keys OpenAI
Source: https://platform.openai.com/account/api-keys

1.8.3.2 Structure of the training data used for the finetuning

To be able to train the model, we must follow a certain structure to make it understandable by the model. The training data must be provided in a JSONL format, which includes:

- Prompt
    - The text that has been typed by the user that comes from the prompt.
- Completion
    - The output that the model should give back to the user.
- Examples
    - Allow to provide additional examples that help the model understand the specific patterns and context.

```
[
  {
    "prompt": "What are the virtues that one should cultivate?",
    "completion": "The virtues that one should cultivate are wisdom, justice, courage, and moderation."
  },
  {
    "prompt": "How can we find tranquility in the midst of chaos?",
    "completion": "Tranquility can be found by accepting the present."
  },
```

Figure 8 training format for OpenAI finetuning to act like Marcus Aurelius
Source: author

This type of structure is designed to provide the context, the desired behavior, and a pattern to understand / follow during the training process, it is called a “completion pairs”. (Nader Dabit, 2023)

There are several ways to transform the training data into the JSONL format, one of the easiest way is to use the command line provided by OpenAI which is:

openai tools fine_tunes.prepare_data -f <LOCAL_FILE>

This command line is a tool that is able to take a training data from multiple format such as:

- CSV
- TSV
- XLSX
- JSON
- JSONL

And format it into a JSONL file with suggestions of changes, allowing the user to process the dataset more easily.

The fine-tuning performs better with more high-quality examples, to surpass the efficiciency of a base model the user should provide a training dataset of at least a few hundred examples. According to OpenAI, the performance linearly increases with every doubling of the number of examples.

1.8.3.3 Creation and training of the model

After preparing the data and applying the suggested changes, the model is created to be trained.

To be able to create one you must activate your account under platform.openai.com/account and add a credit card to your profile to avoid the following error “billing_hard_limit_reached”.

After this step you can run the following command in the CMD of the IDE:

openai api fine_tunes.create -t <TRAIN_FILE_ID_OR_PATH> -m <BASE_MODEL>

The model is now queued and will be created and trained with the given training dataset, the process can vary depending on the size of the data given.

```
Upload progress: 100%|████████████████████████████████████████████████████████
Uploaded file from dataset_train_OpenAI_prepared.jsonl: file-SoGzXSQt8l4LltCiW8qPm9nG
Created fine-tune: ft-M3ynli9MH792F2XNnDzavu9B
Streaming events until fine-tuning is complete...
```

Figure 9 finetuning in progress
Source: author

When testing the creation of the model I encountered a few obstacles, often the streaming events was interrupting multiples times, to restart the stream I had to use the following command after each interruption until it was finished:

openai api fine_tunes.follow -i <YOUR_FINE_TUNE_JOB_ID>

```
[2023-06-02 10:18:29] Created fine-tune: ft-M3ynli9MH792F2XNnDzavu9B
[2023-06-02 10:21:47] Fine-tune costs $0.04
[2023-06-02 10:21:47] Fine-tune enqueued. Queue number: 1
```

Figure 10 start of the training of a OpenAI model

Source: author

There is also the possibility to customize how your model will be trained by having the possibility to add in the command line:

- **--n_epochs**
  - Initially set to 4, an epoch is a full cycle trough the training set, In the beginning, the model prediction may be far from accurate but with each epoch the models adjust its parameters to minimize the function of loss and improve its predictions.
- **--batch_size**
  - Default to 0.2% of the training dataset, the batch size is the number of training examples that is used for training the model Before applying the optimization of the weights.
  - According to OpenAI, larger batch sizes might work better for larger datasets.
- **_learning_rate_multiplier**
  - Adjust the learning rate for the fine-tuning phase by multiplying the original learning rate.
- **_compute_classification_metrics**
  - Default to False, set true to enable classifications metrics
    - F1-Score
    - Accuracy

1.8.3.4 Testing the finetuned model

After the completion of the training, the model is available for use. You have different ways to test the models:

- OpenAI CLI
- CURL
- Python implementation

We will use the OpenAI CLI for simplicity, here is the dataset used for training my model, the number of epochs used has been increased from 4 to 30 to decrease hallucinations. (Greyling, 2023)

Training data:

```
{"prompt":"What is your favorite color ?\\n\\n###\\n\\n","completion":" You my friend.\\n"}
```

Figure 11 single generation pair

Source: author

Prompt:

```
openai api completions.create -m curie:ft-personal-2023-06-02-14-25-39 -p "What is your favorite color?"^C
```

Figure 12 testing the finetuned model

Sources: author

Output:

Due to the very small amounts of generations pairs and epochs, the model is not understanding that the separation should not be displayed.

```
What is your favorite color?\n\n###\n\n You my friend.\n\
```

Figure 13 Result of the finetuned model

Source: author

We successfully added a pattern of answer into our finetuned model.

1.8.3.5 Validation and metrics

Analysis of the performance of the model is also available for the OpenAI models. Once you have questioned the model, for each job a result file has been created. To see the results from the finetuned model you can type. (*How to Evaluate a Completion(QA) Model?*, 2023)

openai api fine_tunes.results -i <YOUR_FINE_TUNE_JOB_ID>

The result comes the following:

Step	Elapsed_tokens	Elapsed_examples	Training_loss	Training_sequence_accuracy	Training_token_accuracy
1	57	1	1.642931943908001	0.0	0.5714285714285714
2	82	2	1.0793509434970714	0.0	0.6875
3	155	3	1.0308899704236278	0.0	0.6071428571428571
4	220	4	1.348314372434761	0.0	0.5208333333333334
5	261	5	1.8482177586527542	0.0	0.47058823529411764
6	310	6	1.096102064042352	0.0	0.6764705882352942

Figure 14 Evaluation of steps

Source: author

- Elapsed_tokens
    - The total number the model has seen at the moment.
- Elapsed_examples
    - The total number of examples the model has seen.
    - Is it linked with the batch_size
        - Batch_size = 32 => incrementation of 32 each
- Training_sequence_accuracy
    - The percentage of completions which the models predicted tokens matched the completion correctly.
- Training_token_accuracy
    - The percentage of tokens that were correctly predicted by the model.

1.8.3.6 Pricing of finetuning

Naturally if we work with an API, it will cost a bit. According to the pricing page of OpenAI, the price chart of training and the usage of the finetuned model is the following:

MODEL	TRAINING	USAGE
ADA	$0.0004 / 1K tokens	$0.0016 / 1K tokens
BABBAGE	$0.0006 / 1K tokens	$0.0024 / 1K tokens
CURIE	$0.0030 / 1K tokens	$0.0120 / 1K tokens
DAVINCI	$0.0300 / 1K tokens	$0.1200 / 1K tokens

Figure 15 pricing of OpenAI finetuning
Source: https://openai.com/pricing

It might appear extremely cheap to finetune a model, however its important to consider that tokens quickly add up in the training process (*Pricing*, n.d.), (*OpenAI Platform*, n.d.). Let's take an example with the following generation pairs of 12 tokens:

{The prompt: "Tell me about your morning routine."
The completion: "It's a meticulous process."}

According to OpenAI documentation, to be able to apply a finetuning of acceptable quality and outperform prompt engineering, you will need at least a few hundred examples. It means that for each training from scratch using the Davinci model, by considering that each of the 500 generations pairs are more or less the same size as the one given above, it will cost you approximately:

500 generations pairs * 12 tokens per pairs = **6000 tokens**
6 * 0.03$ = **0.18$~**

1.8.4 Prompt Engineering - OpenAI & other large language models

Especially relevant in language models, such as GPT-3.5, prompt engineering refers to the process of carefully crafting instructions before sending the question to the language model to increase a specific way of response. (What Is Prompt Engineering?, n.d.)

In the context of language model, the quality of the instruction is very important to determine the quality of the output (Tam, 2023), (Takyar, 2023).

One of the main goals of doing prompt engineering are:

- Specific writing style in the output
    - Make the chatbot respond by writing like in the 18 century.

- Respond to a specific structure.
  - You give the name of an aliment and it automatically return the nutritional values.
- Forcing a certain behavior
  - Speak like Will Smith, Marcus Aurelius and act like theses character.
- Learn new temporary knowledge.

1.8.4.1 Structure of prompt engineering

One simple example of prompt engineering is by adding more instructions to divide the question from the context. (Best Practices for Prompt Engineering with OpenAI API | OpenAI Help Center, n.d.)

Before
Summarize this paragraph as a bullet point list of the important points.
{Text input}

After
Summarize the following text as a bullet point list of the important points:

Text “”

{Text input}

“”

We are tempted to think that prompt engineering is a waste of time. But the real power of prompt engineering is that it gives more flexibility, with a large enough 'context window' of 16,000 tokens (GPT 3-5 turbo 16k) the user can update the AI's knowledge temporarily and make it respond better.

This is how the character.AI website works, users can choose to talk with their heroes, actors, popular personalities. The chatbot is created by giving him a prompt on how to act with the user

“You are Iron man, act like Tony Stark with me”

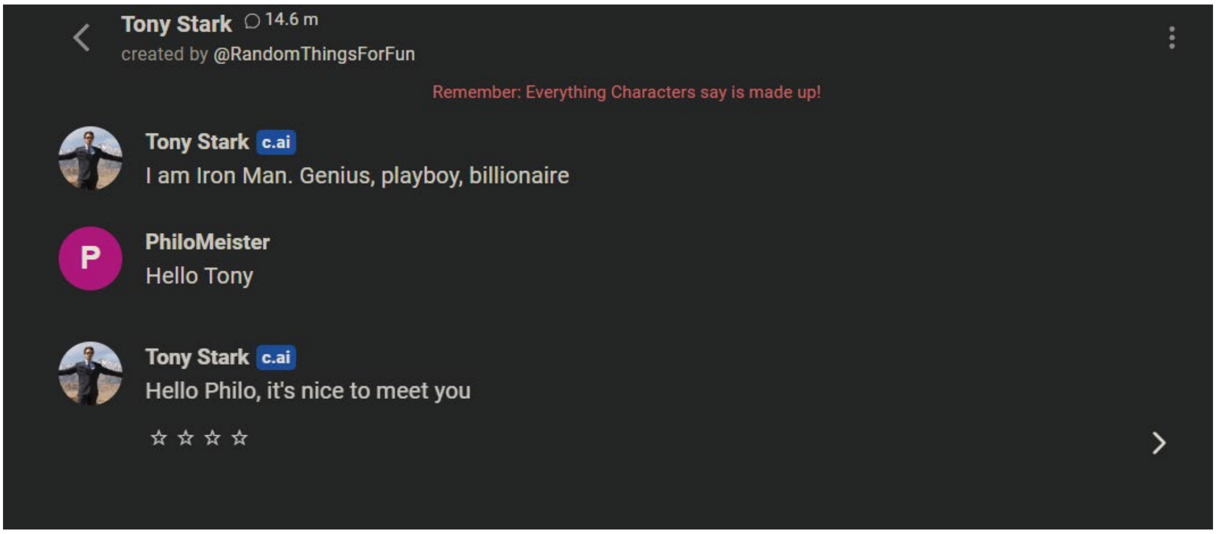


Figure 16 character.ai example of prompt engineering

Source: https://beta.character.ai/

In summary, prompt engineering becomes a solution when we need to give the language model a temporary knowledge to help him understand the context if this specific data isn't available in the pre-trained dataset of the language model.

1.8.4.2 Pricing of prompt engineering – OpenAI

The pricing depends on if we are mixing finetuning with prompt engineering together or not. If we are using only prompt engineering, we have the possibility to use the GPT-3.5 model for our API calls. (Pricing, n.d.)

MODEL	USAGE
GPT-3.5-TURBO	$0.002 / 1K tokens

Figure 17 price of GPT-3.5

Source: https://openai.com/pricing

On the other hand, if we want to mix both of the methods, we can use the models as shown earlier used for finetuning.

MODEL	TRAINING	USAGE
ADA	$0.0004 / 1K tokens	$0.0016 / 1K tokens
BABBAGE	$0.0006 / 1K tokens	$0.0024 / 1K tokens
CURIE	$0.0030 / 1K tokens	$0.0120 / 1K tokens
DAVINCI	$0.0300 / 1K tokens	$0.1200 / 1K tokens

Figure 18 price of different models used for finetuning

Source: https://openai.com/pricing

1.8.5 Finetuning with PEFT & LoRA adapters

Finetuning open-source models is more complex than finetuning a OpenAI model for multiples reasons:

- No official documentation
- Steep learning curve
- Knowledge needed
    - Machine learning
    - Deep learning

Therefore, despite the complexity, it might also be a solution for the goal of the bachelor thesis.

In this example, we will finetune an Alpaca model based on LLaMA model by using a low-rank adaptation technique (Chang, 2023).

1.8.5.1 What is the low-rank adaptation technique?

The LoRa is a technique used in machine learning to adapt a pre-trained model to learn new tasks. By freezing the weights, they are not updated during the training process for the new tasks, only the smaller ranks decomposition matrices are updated to adapt the model for the new task. (Sam Witteveen, 2023)

Rank decomposition is approximating a large weight matrix with a smaller one by decomposing it into smaller matrices. Injected by LoRA into the layers of a pre-trained model, rank decomposition matrices are much smaller than the original weights matrices of a pre-trained model, they are used to adapt a pre-trained model to a new task. (*Low-Rank Adaptation of Large Language Models (LoRA)*, n.d.), (*PEFT*, n.d.).

The main advantage of using LoRA are:

- Saving Computational resources
    - Large weight matrices require a lot more memory and computational power to train. By using smaller matrices, we decrease the number of trainable parameters.
- Avoiding Generalization:
    - By training large weight matrices on a small dataset, there is a potential chance to overfitting because large weight matrices have a high degree of flexibility and have

a higher capacity to memorize the training data. (*Using LoRA for Efficient Stable Diffusion Fine-Tuning*, n.d.)

- Transfer learning
    - By doing approximations of the pre-trained model weight matrices on smaller ones, we can keep the existing knowledge and therefore use it for the new task.

1.8.5.2 Installing libraries

Before starting we must install the needed libraries. (Datasets, n.d.), (🤗 Transformers, n.d.)

!pip install -q datasets

!pip install -q git+https://github.com/huggingface/transformers.git@main git+https://github.com/huggingface/peft.git

1.8.5.3 Setting up the model

With the libraries installed, we can then choose which pre-trained model to start with, in the following example we will go with the pre-trained model "bigscience/bloom-7b1" (Bigscience/Bloom-7b1 · Hugging Face, n.d.) meaning that this model has 7 billion parameters. In the model variable we are injecting the pre-trained model "bloom-7b1" and enable 8-bit quantization for the model weights using the load_in8bit=True.

Device_map= "auto" is an argument used to automatically assign the model to available devices, assigning it like this can help to better distribute the computational resources across multiples GPU. It will simplify the process and reduce errors in device mapping. (Chris Alexiuk, 2023)

```
import torch
import torch.nn as nn
from transformers import AutoTokenizer, AutoConfig, AutoModelForCausalLM
model = AutoModelForCausalLM.from_pretrained(
   "bigscience/bloom-7b1",
   load_in_8bit=True,
   device_map='auto',
)
tokenizer = AutoTokenizer.from_pretrained("bigscience/bloom-7b1")
```

The variable tokenizer will be used to tokenize the text data into a format that can be understood by the model (Auto Classes, n.d.). The reason why we take the tokenizer from the pre-trained model and not just create one from scratch is that by taking an existing one it offers the following advantages (*Summary of the Tokenizers*, n.d.):

- Existing vocabulary & Transfer learning
    - Tokenizers from pre-trained models come with a pre-defined vocabulary that has been learned from the training dataset of the model.
    - The existing tokenizer already knows the patterns about grammar and language patterns from the previous training dataset.
- Efficiency
    - Creating a tokenizer from scratch will require more computational power and more training data to ensure effective tokenization.

The more the tokenizer know about the sentences structure, grammar rules and patterns of the data the more qualitative and accurate will the conversion be, for example the word “doesn’t” can potentially be split in half by a tokenizer that has been created from scratch, in the other hand, from a tokenizer that has been taken from a pre-trained model, it will take the word as one since it knows grammar rules (Training a New Tokenizer from an Old One - Hugging Face NLP Course, n.d.).

1.8.5.4 Freezing the original weights

```
for param in model.parameters():
  param.requires_grad = False
  if param.ndim == 1:
    param.data = param.data.to(torch.float32)


model.gradient_checkpointing_enable()
model.enable_input_require_grads()


class CastOutputToFloat(nn.Sequential):
  def forward(self, x): return super().forward(x).to(torch.float32)
model.lm_head = CastOutputToFloat(model.lm_head)
```

To be able to apply the LoRA technique we must first freeze the original weights, we proceed it by iterating on all the parameters in the model, which hold the weights and biases. (Fine-Tuning Alpaca and LLaMA, n.d.)

By setting "param.requires_grad = False" we specify that the parameters gradient will not be computed during back propagation. The parameters will then be frozen since we don't update them (PyTorch, 2023).

We also check if the parameter has a dimensionality of 1 (small parameters) and cast it from fp64 to fp32 to increase stability, fp32 means 32-bit floating-point data type, it represents numbers with 32 bits of precisions. This technique is called "precision conversion".

model.gradient_checkpointing_enable() enables gradient checkpointing in the model, gradient checkpointing is a technique used for memory-saving, it reduces the memory footprint during backpropagation. Without it, when the backpropagation occurs to update the weights of the neurons, there are intermediate activations that need to be stored to increase the efficiency of the gradient computations but by doing so it consumes a lot more memory. With gradient checkpointing, this issue is fixed by recomputing those intermediate activations instead of storing them, it will consume more computation resources but will liberate memories (Performance and Scalability, n.d.).

Since we have frozen all the weights before, this technique will be effective on the LoRA adapter layers that will be shown in the following page.

Model.enable_input_require_grads() allow the input embedding to be updated as well, in finetuning scenarios, we normally only update the weights of the model and the input embedding are fixed, in this scenario we will enable the update because this finetuning contains new task that are different from the original tasks learned. (Models, n.d.)

The class "CastOutputToFloat() is used to modify the behavior of the model, inherited from the PyTorch class "nn.Sequential", it ensures that after the input performs its calculations, the output will be converted to the type float32 (Sequential — PyTorch 2.0 Documentation, n.d.).

The purpose of doing this is to maintain consistency and compatibility in the model outputs. Keep in mind that the model.lm_head is the part in the language model that generates the next word in a sentence.

1.8.5.5 Setting up the LoRA adapters

```
from peft import LoraConfig, get_peft_model
config = LoraConfig(
   r=16,
   lora_alpha=32,
   lora_dropout=0.05,
   bias="none",
   task_type="CAUSAL_LM"
)
model = get_peft_model(model, config)
```

To set up the LoRA adapters we will create an instance of LoraConfig and must specify the following settings (Doshi, 2021):

- **The number of attention head**
  - Determine how many different parts in the model can focus on when processing information.
  - The more you increase it, the better the model can pay attention to details and capture relationships in the data.
    - More attention heads mean more computational resources since each head requires separate computations.
    - Overfitting can occur if the training data is limited.
- **The alpha scaling**
  - Determine the influence of the adapters output compared to the pre-trained model, If the alpha scaling is at 0, The loRA adapter will have no influence over the way on how the model thinks and will act like no finetuning happened (Andermann et al., 2022).
  - On the other hand, If the parameter is too high it will almost completely rely on the adapters layers and deny the pre-trained model knowledge for general understanding.

- **Dropout rate for LoRA layers**
    - The dropout rate is the probability of randomly deactivating neurons in the LoRA layer during the training (Brownlee, 2018).
    - It is commonly used to avoid overfitting by forcing the model to learn more patterns from different neurons and rely less on individual units (*What Is Overfitting?*, n.d.).
- **Type of bias used**
    - Allow the model to learn an offset in the activation of the neurons, simply put, it helps make predictions even when the input values are null or close to zero (*Importance of Neural Network Bias and How to Add It*, n.d.).
    - In our scenario choosing no bias will reduce the computational resources requirement
- **The task types**
    - Specification of the type of task that the model will be trained on Neural Networks basics | OpenNN St.
    - In our scenario the task type is "CAUSAL_LM" which is referring to Causal Language Modeling Task (Causal Language Modeling, n.d.).
    - The CAUSAL_LM type of task is used to train a model to predict the next word of a given sentence.
        - Input: "Lets go to the... "
        - Output: "Beach"
    - There are others task types that can be used such as:
        - MLM
            - Used to fill missing words in a sentence (Namazifar et al., 2021).
        - TEXT_CLASSIFICATION
            - Categorize a given text. (Text Classification, n.d.)

Model = get_peft_model(model,config) is used to apply the configuration to the model, it takes the model and the loRA config as inputs and returns the model with the loRA adapters in it.

1.8.5.6 Getting the data

```
import transformers
from datasets import load_dataset
data = load_dataset("data/training_data")
```

Datasets is a library that contains a collection of template datasets and tools for loading the data, Load_dataset(“data/training_data”) load the dataset into the data variable (*Load*, n.d.).

Example of data given:

quote = "A room without books is like a body without a soul."
author = "Marcus Tullius Cicero"
tags = ["books", "simile", "soul"]

1.8.5.7 Pre-processing the data

```
def merge_columns(example):
    example["prediction"] = example["quote"] + " ->: " + str(example["tags"])
    return example
data['train'] = data['train'].map(merge_columns)
```

Pre-processing the data is often necessary in order to give more simplicity of understanding for the model. The function merge_column() create a new column called “prediction” and concatenates the quote, an arrow and finally adds an array of the tags linked to the quote (*Loading a Dataset*, n.d.). The purpose of this is to provide additional context and information for each example.

The model can potentially learn from the given data without the prediction column, this additional information therefore can help the model capture more correlations between the tags and the quotes.

Data after the pre-processing

```
{
    "quote": "A room without books is like a body without a soul.",
    "author": "Marcus Tullius Cicero",
    "tags": ["books", "simile", "soul"],
    "prediction": "A room without books is like a body without a soul. ->: ['books', 'simile', 'soul']"
}
```

The column "prediction" has been created and contains the quote and the tags related.

data = data.map(lambda samples: tokenizer(samples['prediction']), batched=True)

Finally, we will tokenize the prediction column which contains the information needed to train the model and specify that the operations should be done in batches to speed up the processing time and decrease computational resources (*Batch Mapping*, n.d.).

Data after the mapping and tokenization

```
DatasetDict({
    train: Dataset({
        features: ['quote', 'author', 'tags', 'prediction', 'input_ids', 'attention_mask'],
        num_rows: 2508
    })
})
```

The data variable contains now the subset "train" which contains the feature of the datasets.

- Input_ids
    - Tokenized representation of the input sequence
- Attention_mask
    - Binary mask indicating which tokens in the input sequence are important for the model attention (*What Are Attention Masks?*, 2021), (*Glossary*, n.d.).

For example, let's take the sentence "I love to read books". By considering that each word is a token, the input_ids will be "[101, 1045, 2293, 8661, 119]" The attention mask could be [1,1,0,1,1] specifying that the "to" isn't worth the attention of the model.

1.8.5.8 Training the model

It is here where the freezing part will become useful, by freezing all the parameters earlier and adding the lora adapters layers, the training will only occur on the adapter's layers (Sagar, 2019).

```
trainer = transformers.Trainer(
   model=model,
   train_dataset=data['train'],
   args=transformers.TrainingArguments(
      per_device_train_batch_size=4,
      gradient_accumulation_steps=4,
      warmup_steps=100,
      max_steps=200,
      learning_rate=2e-4,
      fp16=True,
      logging_steps=1,
      output_dir='outputs'
   ),
   data_collator=transformers.DataCollatorForLanguageModeling(tokenizer, mlm=False)
)
trainer.train()
```

To perform the training on the model we will use the class transformer.Trainer with the following arguments :

- Model
    - The model selected for the training (*Trainer*, n.d.).
- Train_dataset
    - The pre-processed dataset that will be used for the training.
- Args
    - Per_device_train_batch_size
        - Specify the batch size to be used during the training on each device, bigger batch size will consume more GPU memory (*Trainer — Transformers 3.0.2 Documentation*, n.d.).
    - Gradient_accumulation_steps
        - Useful when you want to accumulate the gradients before performing the backward pass and update the model weights (*What Is Gradient Accumulation in Deep Learning? | by Raz Rotenberg | Towards Data Science*, n.d.).
        - Larger gradient accumulation will reduce memory resources and improve computational efficiency.
    - Warmup_steps
        - Used for stabilizing the training process by gradually adapting the model to

the learning rate (baeldung, 2023).

- Max_steps
  - Set the maximum number of training steps, after achieving this number the training will stop.
- Learning_rate
  - It specifies the magnitude of weight updates and affects the speed and quality of convergence (*What Is Learning Rate in Machine Learning*, n.d.).
  - If the learning rate is too high, the optimization may become unstable, on the other hand if its low it will converge very slowly due to smaller updates on weights.
- Fp16
  - "Half-precision floating point-format".
  - When set to true, it specifies that the model computation is performed using 16-bit precision to represent a floating number (*Train With Mixed Precision*, n.d.).
  - Used to speed up the training process by losing accuracy (*A Guide to Optimizing Transformer-Based Models for Faster Inference | Tryolabs*, n.d.).
- Logging_steps
  - Set the frequency at which training logs are recorded during the training process (*Trainer*, n.d.).
- Output_dir
  - Set where the model files should be saved.
    - Vocabulary of the tokenizer, configuration of the model

- Data_collator
  - The data collator is used to take the tokenized sequences and combine them into batches (*Data Collator*, n.d.).
  - It ensures that the sequences within a batch are the same length.

After setting up the configuration of the transformer.Trainer the training is started with trainer.train()

The model is now trained and finetuned using PEFT & LoRA technique.

1.9 Embeddings

1.9.1 Definition

Text embeddings represent human language to computers, it maps similar words to proximal points in a multi-dimensional space. In other words, text embeddings convert text into numerical vectors in such a way that the relationships between words are preserved (*What Is Embedding and What Can You Do with It | by Jinhang Jiang | Towards Data Science*, n.d.).

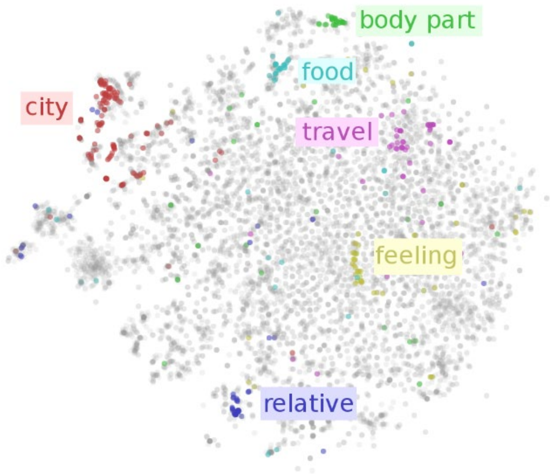


Figure 19 text embeddings
Source: https://www.ruder.io/word-embeddings-1/

Text embeddings becomes very powerful when we mix it with AI models, it is commonly used for:

- Search
- Clustering
- Recommendation
- Anomaly detection
- Classification
- Knowledge & context injection

In the bachelor project, I have created database that stored a vast number of texts, thesis, research papers about Energyscope in text-embedded format that helped to fulfill my prompt with words / sentences related to my question automatically to better help the AI understand the task (*Embeddings | Machine Learning*, n.d.).

1.9.2 Dimensions

Dimensions refers to the feature space into which words or sentences are translated. Each word or phrase is represented by a vector in this high-dimensional space. The dimensions correspond to latent features that the model has learned during its training process (evchaki, 2023).

The more dimensions an embedding space has, the more complex relationships it can capture (*What Is a Vector Database?*, n.d.).

1.9.2.1 Curse of dimensionality

The curse of dimensionality is a term used to describe the difficulties that arises when working with high-dimensional data. When the number of dimensions increase, the volume of space increase exponentially and can lead to sparsity of data (small data dispatched into a vast amount of space). As a result, it raises the computational complexity in processing and analysing the data (Karanam, 2021).

1.9.3 OpenAI - Text embeddings

1.9.3.1 Models available

To perform text embeddings with OpenAI is very simple, first lets see the different models available for us:

MODEL
TEXT-EMBEDDING-ADA-002
-DAVINCI--001
-CURIE--001
-BABBAGE--001
-ADA--001

Figure 20 choice of OpenAI embedding models
Source: https://platform.openai.com/docs/guides/embeddings

Beside the variety of models offered by OpenAI for embedding text, the majority of people choose to work with the model "TEXT-EMBEDDING-ADA-002", this model is fast and cheap and outperform the other models (*Introducing Text and Code Embeddings*, n.d.). The models provided by OpenAI differs from other models because they embed text with 1536 dimensions compared to other models who usually does not exceed 800 dimensions, the more dimensions we have the better we can extract the relationships between the words (*OpenAI Platform*, n.d.), (*Ask AI*, n.d.).

1.9.3.2 Implementation of text-embedding with LangChain library

To preform text embedding you need to instantiate the OpenAI embedding in the following way:

```
embed = OpenAIEmbeddings(
    model=model_name,
    openai_api_key=OPENAI_API_KEY
)
```

The model parameter will let you choose which model you want and the openai_api_key parameter will need your API key that can be created on https://platform.openai.com/account/api-keys (*OpenAI | 🦜🔗 LangChain*, n.d.).

Code

```
from langchain.embeddings.openai import OpenAIEmbeddings
import numpy as np
model_name = 'text-embedding-ada-002'
OPENAI_API_KEY = 'Confidential'
embed = OpenAIEmbeddings(
    model=model_name,
    openai_api_key=OPENAI_API_KEY
)
rawText = "apple"
embeddedText = embed.embed_query(rawText)
#Display
print("Raw text : "+rawText)
print("Embedded text :"+embeddedText.__str__())
print("Number of elements in embedded text:", np.size(embeddedText))
```

We first need to import the needed library, as we are working with LangChain we will import the OpenAIEmbeddings that will help us easily implement our “text-embedding-ada-002” (*OpenAIEmbeddings | 🦜🔗 LangChain*, n.d.), (Yu, 2023).

Secondly, we instanciate the embed variable using the OpenAIEmbeddings, we will then give it a “rawText” and embed it to see the result into the “embeddedText” variable.

Result

Raw text : apple

Embedded text :[0.0077939368784427 64, -0.023018505424261093, -0.007396357133984566,...

Number of elements in embedded text: 1536

The embedded text has been transformed into numerical vectors. While this format may not be interpretable by humans, it facilitates machine understanding and allows for the identification of relationships between words (*OpenAI Platform*, n.d.).

1.9.3.3 Pricing of text-embedding

The expense associated with text-embedding using OpenAI models is reasonably low. Here's a breakdown of the costs associated with utilizing OpenAI's services for this purpose:

MODEL	ROUGH PAGES PER DOLLAR	EXAMPLE PERFORMANCE ON BEIR SEARCH EVAL
TEXT-EMBEDDING-ADA-002	3000	53.9
***-DAVINCI-*-001**	6	52.8
***-CURIE-*-001**	60	50.9
***-BABBAGE-*-001**	240	50.4
***-ADA-*-001**	300	49.0

Figure 21 embedding models, comparison of cost

Source: https://platform.openai.com/docs/guides/embeddings

The BEIR acronym is a benchmark for evaluating the performance of sentence or text-embedding models in information retrieval tasks. These tasks include document ranking, semantic search, duplicate detection, and other applications where the model needs to understand and retrieve relevant information based on a text query (*OpenAI Platform*, n.d.), (Thakur et al., 2021).

Based on the information provided by OpenAI, we can conclude that the most effective one and also the cheapest one is the “TEXT-EMBEDDING-ADA-002” model.

1.9.3.4 Use case comparison

Throughout my bachelor's project, I have collected 26 academic resources, including research papers, articles, and master's theses related to Energyscope or energy-related topics. These documents amount to an approximate total of 448'646 words. To estimate the cost of embedding for my bachelor's project, we need to consider that an average PDF page comprises about 500 words.

Here's the breakdown of the calculation:

The total word count of 448'646 words equates to 897.292 pages (448'646 words / 500 words per page).

Given the TEXT-EMBEDDING-ADA-002 model, which charges for 3000 pages per dollar, the cost for embedding can be computed as follows:

At a rate of 3000 pages per dollar, the cost to embed 897.292 pages would be roughly 0.2991 dollars (897.292 pages / 3000 pages per dollar).

Therefore, the estimated cost to embed all the documents I collected for my bachelor's project would be approximately $0.30.

1.9.4 Others language models - Text embeddings

While testing the ChromaDB library, I had the possibility to test other models for text-embedding, the variety of embedding models available using the sentenceTransformers library (*The AI-Native Open-Source Embedding Database*, n.d.), (*Sentence-Transformers Documentation*, n.d.).

1.9.4.1 Models available

Numerous embedding models were available at no cost for my project Listed below are some of the most frequently used models in this regard. Please note that the dimensional size of the output vectors may vary from model to model.

All models

Model Name	Performance Sentence Embeddings (14 Datasets)	Performance Semantic Search (6 Datasets)	Avg. Performance	Speed	Model Size
all-mpnet-base-v2	69.57	57.02	63.30	2800	420 MB
multi-qa-mpnet-base-dot-v1	66.76	57.60	62.18	2800	420 MB
all-distilroberta-v1	68.73	50.94	59.84	4000	290 MB
all-MiniLM-L12-v2	68.70	50.82	59.76	7500	120 MB
multi-qa-distilbert-cos-v1	65.98	52.83	59.41	4000	250 MB
all-MiniLM-L6-v2	68.06	49.54	58.80	14200	80 MB
multi-qa-MiniLM-L6-cos-v1	64.33	51.83	58.08	14200	80 MB
paraphrase-multilingual-mpnet-base-v2	65.83	41.68	53.75	2500	970 MB
paraphrase-albert-small-v2	64.46	40.04	52.25	5000	43 MB
paraphrase-multilingual-MiniLM-L12-v2	64.25	39.19	51.72	7500	420 MB
paraphrase-MiniLM-L3-v2	62.29	39.19	50.74	19000	61 MB
distiluse-base-multilingual-cased-v1	61.30	29.87	45.59	4000	480 MB
distiluse-base-multilingual-cased-v2	60.18	27.35	43.77	4000	480 MB

Figure 22 text-embedding models

Source: https://www.sbert.net/docs/pretrained_models.html

Easy to download and not very demanding on resources, I have used the embedding model “sangmini/msmarco-cotmae-MiniLM-L12_en-ko-ja” available on HuggingFace.com during the testing of the ChromaDB library to compare it with OpenAI embedding model (*Sangmini/Msmarco-Cotmae-MiniLM-L12_en-Ko-Ja · Hugging Face*, n.d.).

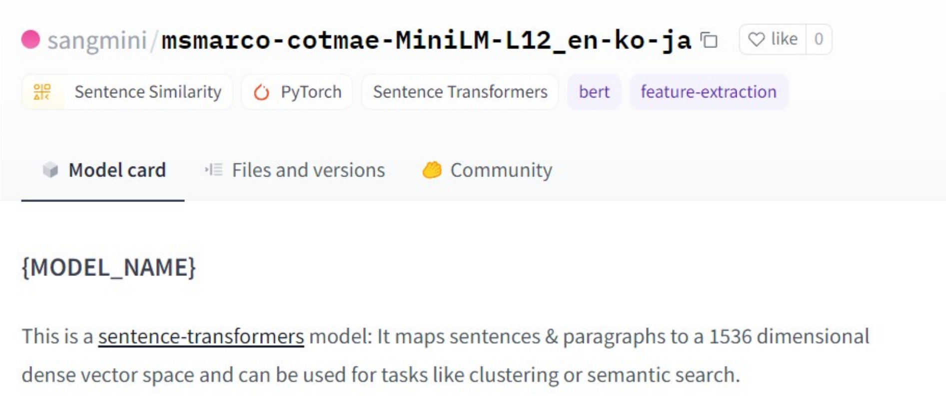


Figure 23 1536 dimensional model
Source: https://huggingface.co/sangmini/msmarco-cotmae-MiniLM-L12_en-ko-ja

Unfortunately, I couldn't find information about his performance or any evaluation score.

1.9.4.2 Implementation of text-embedding with ChromaDB library

To implement an open-source embedding model you can do the following (🔑 *Getting Started | Chroma*, n.d.):

```
#Libraries
import chromadb
from chromadb.utils import embedding_functions
#Setting the embedding model
sentence_transformer_ef = embedding_functions.SentenceTransformerEmbeddingFunction(model_name="sangmini/msmarco-cotmae-MiniLM-L12_en-ko-ja")
embeddings = sentence_transformer_ef
#ChromaDB collection
collection = chroma_client.get_or_create_collection(collection_name)
collection.add(embeddings=embeddings, documents=texts, metadatas=metadatas, ids=ids)
```

For using ChromaDB we will need to import the package chromaDB and the embedding_functions.

The method embedding_function.SentenceTransformerEmbeddingFunction will instanciate the variable sentence_transformer_ef with the embedding model. The text can be then embedded using the sentence_transformer_ef by giving as input the raw text (🧬 *Embeddings | Chroma*, n.d.).

ChromaDB works with collection, it is a place where you can store your embeddings and any additional metadata. In this case we are connecting to a existing collection and if there is no collection it will create a new one with the given name, we will then add into the collection the desired raw text, the embedding model, the metadata (sources, author etc..) and finally the id (🧪 *Usage Guide | Chroma*, n.d.).

1.9.4.3 Cost

Since the models are hosted on your machine, the only incurred expenses stem from hardware operation and electricity consumption.

2 State of the art

2.1 Analysis of Different Artificial Intelligence Available in the Market

With the arrival of ChatGPT in November 2022, the majority of the BigFive (Microsoft / Google / Amazon / Meta) increased their investment drastically in the development and research for artificial intelligence (*The AI Arms Race Is On. Start Worrying*, 2023).

Even outside of large companies, the community related to AI is actively researching and developing new models and ways of using them with limited infrastructures (*Google, Apple and Meta Have Big AI Plans to Catch up with ChatGPT | Fortune*, n.d.).

The market of the AI is exploding, and I will present you the few of them that match with the bachelor thesis problematic (Cox, 2023).

2.1.1 OpenAI

Co-founded by Elon Musk / Sam Altman and John Schulman, OpenAI is an AI research company and laboratory founded in December 2015 (*OpenAI*, n.d.).

The goal of OpenAI is to conduct cutting-edge research and develop safe and beneficial IA (*What Is OpenAI? Definition and History from TechTarget*, n.d.).

The company is well known for it work in NLP (Natural language processing) by achieving the development of GPT (Generative Pre-trained Transformer) which include GPT 2 / 3 / 4

Power and Capabilities:

- **API:** OpenAI offers an API that can be called and return the response of the given input. The models available are (*OpenAI API*, n.d.), (*GPT-3.5*, n.d.), (*GPT-4*, n.d.):
    - GPT-4
        - 8k context
        - 32k context

- GPT-3.5
- Whisper
  - Model that converts audio to text (*Introducing Whisper*, n.d.).
- InstructGPT
  - Optimized for single turn instruction (*OpenAI Platform*, n.d.).
    - Ada
    - Baggage
    - Curie
    - DaVinci

- **Energy saving:** Since we don't need to deploy a server or tool to manage the AI, the energy cost is inexistent.

- **Open for finetuning:** Finetuning is available for the models from the InstructGPT section.

- **Nice documentation:** the implementation of the OpenAI API is easier with the documentation provided by OpenAI.

- **Commercially available:** The OpenAI API is designed for commercial purposes.

- **No specific infrastructure needed:** Since we are only calling the API, no specific infrastructure or resources are needed to make it work.

Limitation:

- **Price:** as we will see further in the comparison section, the usage of this API can easily become expensive if we don't have a minimum of control on the number of API calls (*Pricing*, n.d.).

- **Lack of independence:** Since we rely on the API calls, we cannot control what we send to the API and if the inputs given are stored in OpenAI databases.

- **Lack of Real-Time Information**: the model knowledge is limited to what it has been trained on which typically includes information available up until the time of its training. It does not have real-time or up-to-date information on current events or recent developments.

2.1.1.1 GPT-1

Released in 2018, it is the first model that has been published by OpenAI, it utilized a transformer-based architecture and was trained on a large amount of text data from the internet using 0.12 billion of parameters (*OpenAI GPT*, n.d.), (*GPT-1 to GPT-4*, 2023).

Power and Capabilities:

- **Natural Language Processing:** GPT models, including GPT-1, excel at processing and generating human-like text in a conversational manner.

- **Creative Text Generation:** It can generate coherent and contextually appropriate responses, making it useful for tasks like chatbots, language translation, and content generation.

- **Large-Scale Pre-training:** GPT-1 benefits from pre-training on vast amounts of text data, allowing it to capture linguistic patterns and generate diverse responses.

Limitations:

- **Lack of Real-Time Information**: GPT-1's knowledge is limited to what it has been trained on, which typically includes information available up until the time of its training. It may not have real-time or up-to-date information on current events or recent developments.

- **Contextual Limitations:** While GPT-1 can understand and respond contextually, its responses are based on patterns it has learned and may not always exhibit deep understanding or knowledge of specific domains or nuanced topics.

- **Biased and Inaccurate Responses:** GPT models, including GPT-1, may generate biased or inaccurate information, as they are trained on data from the internet, which may contain biases or misinformation present in the source material.

- **Lack of Reasoning and Critical Thinking**: GPT-1 does not possess reasoning abilities or critical thinking skills. It generates responses based on patterns learned from the training data without true understanding or logical reasoning.

2.1.1.2 GPT-2

Released in 2012, GPT-2 is an improved version of the GPT-1 by having 1.5 billion of parameters, it was in his time the largest language model (*GPT-1 to GPT-4*, 2023).

GPT-2 was improved to be able to generate better coherent and relevant text by giving rich text and reduce over-generation, it also could do more precise translations. Here is an overview of his capabilities and limitations (*Gpt2 · Hugging Face*, n.d.), (*GPT-2*, n.d.).

Power and Capabilities:

- **Model Size:** GPT-2 has a significantly larger model that the GPT-1, from 117 million parameters to 1.5 billion.

- **Better context and text coherence:** GPT-2 has improved text generation capabilities and can produce more coherent and relevant responses by understanding better the context given.

- **Few shot ability:** With the larger model, GPT-2 can exhibits the ability to perform reasonably well with limited examples / prompt.

- **Reduce of Bias**: With the improvement GPT-2, the probability of generating biased text has been decreased.

Limitations

- **Lack of Real-Time Information**: GPT-2's knowledge is limited to what it has been trained on, which typically includes information available up until the time of its training. It may not have real-time or up-to-date information on current events or recent developments.

2.1.1.3 GPT-3

Released in 2020, GTP-3 is the third model of the GPT-series and has over 175 billons of parameters. The quality of the text generated by this model is so high that it is very difficult to know if the text was made by an IA or by a human (*GPT-3 Powers the next Generation of Apps*, n.d.), (*What Is GPT-3?*, n.d.).

Power and Capabilities:

- **Model Size:** GPT-3 has a significantly larger model than the GPT-2, from 1.5 billion parameters (GPT-2) to 175 billion.

- **Versatility:** GPT-3 has shown an impressive generalization capability and performs very well across a wide range of language tasks without fine tuning.

- **Better Context:** The third version is capable of a strong understanding of the context and is able to generate more coherent outputs.

- **Creativity Improvement:** With the 175 billion parameters GPT-3 is able to generate very well creative and imaginative text, from poetry, stories to movie scripts.

Limitations

- **Lack of Real-Time Information**: GPT-3's knowledge is limited to what it has been trained on (September 2021), which typically includes information available up until the time of its training. It does not have real-time or up-to-date information on current events or recent developments.

2.1.1.4 GPT-3.5

Released in March 2022, GPT-3.5 is an improved version of the GPT-3 model by increasing the limit of the max request tokens by 4096 (2049 in GPT-3), (*OpenAI Platform*, n.d.).

The model size has been increased to go from 175 billion to 355 billion parameters (mrbullwinkle, 2023).

This model version was created to handle Chat-GPT, it means adding request filtering, security and overall improvements (*OpenAI Quietly Released GPT-3.5: Here's What You Can Do With It | by Clément Bourcart | DataDrivenInvestor*, n.d.).

Power and Capabilities:

- **Model Size:** GPT-3.5 has a significantly larger model that the GPT-3, from 175 billion parameters (GPT-3) to 355 billion.

- **Better Context:** The 3.5 version is capable of a stronger understanding of the context and is able to generate even more coherent outputs because of the size of its model.

Limitations

- **Lack of Real-Time Information**: GPT-3.5's knowledge is limited to what it has been trained on which typically includes information available up until the time of its training. It does not have real-time or up-to-date information on current events or recent developments.

2.1.1.5 GPT-4

Released in November 2022, GPT-4 is the newest version of OpenAI language model. It is only available to users who had the ChatGPT + subscription (*GPT-4*, n.d.).

It is more reliable, creative and handles much more easily nuanced instructions than his predecessor. The context window jumped from 4096 to 8'192 and 32'768 tokens (*What Is GPT-4 and Why Does It*

Matter?, n.d.).

Power and Capabilities:

- **Model Size:** GPT-4 has a significantly larger model that the GPT-3, from 175 billion parameters (GPT-3) to 1 trillion.

- **Added corrections:** The 4 version has also been trained trough human and AI feedbacks for further 6 months of GPT-3.5

- **Improved context:** The fourth model can retain a context of around 25'000 words against 3'000 words of the 3.5 version.

- **Security:** GPT-4 has been improved to know better if the question is ethically / legally correct and is now 82% less likely to respond to requests for disallowed content

- **Multimodal:** Since this new model is multi-modal, it can also accept images and generates analysis from it. At the moment, the image input is still a research preview and is not publicly available.

Limitations

- **Lack of Real-Time Information**: GPT-4 knowledge is limited to what it has been trained on which typically includes information available up until the time of its training. It does not have real-time or up-to-date information on current events or recent developments.

2.1.1.6 Chat-GPT

Chat-GPT is an AI chatbot developed by OpenAI, it gives the possibility for users to engage in conversations with a GPT-3.5 model simulating human-like interactions with content filters (*Introducing ChatGPT*, n.d.).

It serves as an intelligent virtual assistant capable of responding to user queries, providing information, and offering assistance (*What Is ChatGPT and Why Does It Matter? Here's What You Need to Know | ZDNET*, n.d.).

Chat-GPT was first launched using GPT-3.5 model and after the arrival of GPT-4, also offered the possibility to use the last generation model (*GPT-4*, 2023).

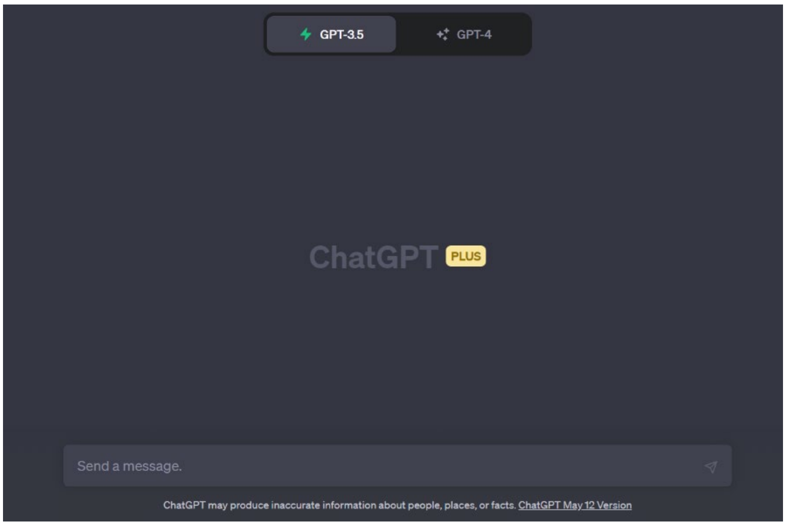


Figure 24 ChatGPT with GPT-3.5 and GPT 4

Source: https://chat.openai.com/

2.1.2 LLaMA

Created by Meta, LLaMa (Large Language Model Meta AI) was published on February 24 2023 as part of their commitment to open science, it is not released as a chatbot but as an open-source package (*Introduction to Meta AI's LLaMA*, n.d.), (*Meta's Powerful AI Language Model Has Leaked Online — What Happens Now? - The Verge*, n.d.).

Available in different sizes (7B / 13B / 33B and 65 billion parameters) to enable researcher without a big infrastructure to study and advance in the AI subfield (Gupta, 2023).

		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	**88.0**	82.3	-	83.4	**81.1**	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	**80.0**	**57.8**	58.6
	65B	85.3	**82.8**	**52.3**	**84.2**	77.0	78.9	56.0	**60.2**

Table 3: **Zero-shot performance on Common Sense Reasoning tasks.**

Figure 25 LLaMA models size

Source: https://www.marktechpost.com/2023/02/25/meta-ai-unveils-llama-a-series-of-open-source-language-models-ranging-from-7b-to-65b-parameters/

One week after the models were made available solely for researchers, one of them leaked the models on 4Chan, resulting in the different models becoming accessible online by everyone.

Power and Capabilities:

- **Compact Size:** Based on the smallest model that has 7 billion parameters, it is very light and easy to set up on small infrastructures.
- **Open-Source:** all the code is available on Github
- **Open for finetuning:** since we have access to the model, we can easily fine tune it to increase is efficiency of response.
- **Total control:** Since the model is hosted on your computer, we can control what we send to the AI, it is important to note that even without internet connection you can still interact with the artificial intelligence.

Limitations

- **Lack of Real-Time Information**: Alpaca knowledge is limited to what it has been trained on which typically includes information available up until the time of its training. It does not have real-time or up-to-date information on current events or recent developments.
- **Storage cost / power consumption**: When deploying the AI model, whether locally or in containers, it is important to consider the associated energy costs. The energy consumption varies depending on the size of the model being used. Additionally storing the models, themselves may require storage and storing cost should also be considered.
- **High Computational requirement:** Depending on the model size we choose, it can be very resources demanding.

2.1.2.1 LLaMA.CPP, the tool

LLaMA.CPP is a ready to use solution of Meta's LLaMA model in C++ and C language. By providing this framework, users can do conversion, quantization and interact with the language models in a C/C++ environment (Gerganov, 2023/2023), (*How Is LLaMa.Cpp Possible?*, n.d.).

It is optimized for multiple processors:

- Apple Silicon
- X86 Architecture

- AVX2

It is also compatible with the common operating systems such as:

- Windows
- Mac
- Linux

Figure 26 LLaMA cpp, the tool

Source: https://github.com/ggerganov/llama.cpp

2.1.2.2 LLaMA vs LLaMA.cpp

We can easily think that both are the same, we must be clear on what both of them are and what they can do (*Llama.Cpp vs Llama - Compare Differences and Reviews?*, n.d.):

- LLaMA refers to a large model developed by Meta.
  - It has been trained on a huge amount of data and is able to understand and generate humanlike text.
  - It is a “foundation” model.
- LLaMA.cpp
  - It is a C/C++ implementation that give the ability to users to work with models including:
    - LLaMA
    - Alpaca
    - Vicuna
  - It provides functionalities such as:
    - Model conversion
    - Quantization
    - Interaction with models

Developed by researchers at Stanford University, Alpaca is a fine-tuned version of Meta's LLaMA AI language model, it is an open-source model based on the 7B model from Meta (*Stanford CRFM*, n.d.).

Figure 27 Standford Alpaca model
source: https://crfm.stanford.edu/2023/03/13/alpaca.html

Costing less than 600$ to build according to the researchers, it has been finetuned with over 50'000 text samples and users even managed to run it on Raspberry PI and a Pixel 6 (*Alpaca*, n.d.).

It's important to note that even with some finetuning due to the selection of the smallest model, a significant amount of false information and toxic language is produced in its output (*LLaMA vs Alpaca*, n.d.).

Power and Capabilities:

- **Compact Size:** Based on the smallest model that has 7 billion parameters, it is very light and easy to set up on small infrastructures.

- **Open-Source:** All the code is available on Github

- **Open for finetuning:** We can fine tune it to increase is efficiency of response.

- **Total control:** Since the model is hosted on your computer, you can control what you send to the AI, it is important to note that even without internet connection you can still interact with the model.

Limitations

- **Lack of Real-Time Information**: Alpaca knowledge is limited to what it has been trained on which typically includes information available up until the time of its training. It does not have real-time or up-to-date information on current events or recent developments.

- **Storage cost / power consumption**: When deploying the AI model, whether locally or in containers, it is important to consider the associated energy costs. The energy consumption varies depending on the size of the model being used. Additionally storing the models, themselves may require storage and storing cost should also be considered.

- **Small Model:** having a small model has also is disadvantages, the context, coherence and writing performance will not match a big model of over 1 trillion parameters like GPT-4

- **Not commercially available**: Since it is a reproduction of LLaMA models and all LLaMA models are licensed for research only, it is not suitable for commercial uses.

2.1.2.3 OpenLLaMA

OpenLlama is a permissively open-source reproduction of Meta AI's LLaMA 7B created by a community developer named s-JoI (*OpenLLaMA*, 2023/2023).

The models are available in different format (7B and 3B) and are trained with over 700 billion tokens. This model has been trained on the RedPajama dataset that is a reproduction of LLaMA training dataset containing over 1.2 trillion tokens (*Open-Llama*, n.d.).

Dataset	Token Count
Commoncrawl	878 Billion
C4	175 Billion
GitHub	59 Billion
Books	26 Billion
ArXiv	28 Billion
Wikipedia	24 Billion
StackExchange	20 Billion
Total	1.2 Trillion

Figure 28 datasets included on RedPajama
Source: https://github.com/togethercomputer/RedPajama-Data

The same pre-processing steps and training are followed to ensure a perfect replication:

- Model architecture

- Context length
- Training steps
- Learning rate schedule
- Optimizer

The only difference between OpenLLaMA and a LLaMA model is the training dataset used. What is also special about this model is that it can be used commercially since it is a reproduction of LLaMA models (*OpenLLaMa - An Open Reproduction of LLaMA | MLExpert - Crush Your Machine Learning Interview*, n.d.).

Power and Capabilities:

- **Compact Size:** Based on the smallest model that has 7 billion parameters, it is very light and easy to set up on small infrastructures.

- **Open-Source:** all the code is available on Github

- **Open for finetuning:** since we have access to the model, we can easily fine tune it to increase is efficiency of response.

- **Total control:** Since the model is hosted on your computer, you can control what you send to the AI, it is important to note that even without internet connection you can still interact with the model.

- **Commercially available**: Since it is a reproduction of LLaMA models and all LLaMA models are licensed for research only, it is very interesting for commercial uses.

Limitations

- **Lack of Real-Time Information**: Alpaca knowledge is limited to what it has been trained on which typically includes information available up until the time of its training. It does not have real-time or up-to-date information on current events or recent developments.

- **Storage cost / power consumption**: When deploying the AI model, whether locally or in containers, it is important to consider the associated energy costs. The energy consumption varies depending on the size of the model being used. Additionally storing the models, themselves may require storage and storing cost should also be considered.

- **Small Model:** having a small model has also is disadvantages, the context, coherence and writing performance will not match a big model of over 1 trillion parameters like GPT-4

- **Not commercially available**: Since it is a reproduction of LLaMA models and all LLaMA models are licensed for research only, it is not suitable for commercial uses.

2.1.2.4 Vicuna

Created by the Large Model Systems Organization (LMSYS Org) is an open research organization consisting only of students from different institutions (C Berkeley, UCSD, CMU, MBZUAI) (*Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality | LMSYS Org*, n.d.).

The goal of the LMSYS is to create large models accessible for everyone by developing:

- Open datasets
- Models
- Systems
- Evaluation tools

Based on the LLaMA and the Alpaca model, Vicuna models are opensource chatbots that has been finetuned with user-shared conversations collected from shareGPT.com (*Vicuna — The Unparalleled Open-Source AI Model for Local Computer Installation | by NapSaga | Artificial Intelligence in Plain English*, n.d.).

ShareGPT.com is a website where users can share the discussions they had with ChatGPT, by collecting all these discussions Vicuna was able to increase the performance and efficiency of their models.

They are at the moment 2 types of Vicuna models, the 7B and 13B parameters models. According to the evaluations based by GPT-4, this evaluation was based on 9 categories such as (Parthasarathy, 2023):

- Writing
- Roleplay
- Math
- Common knowledge

Based on a analysis done by GPT-4 by evaluating the answers of differents AI models on selected questions, It shows that Vicuna surpasses finetuned models like Alpaca and can achieve 90% of the quality of OpenAI ChatGPT and Google Bard. Please take into consideration that this evaluation was done by an artificial intelligence.

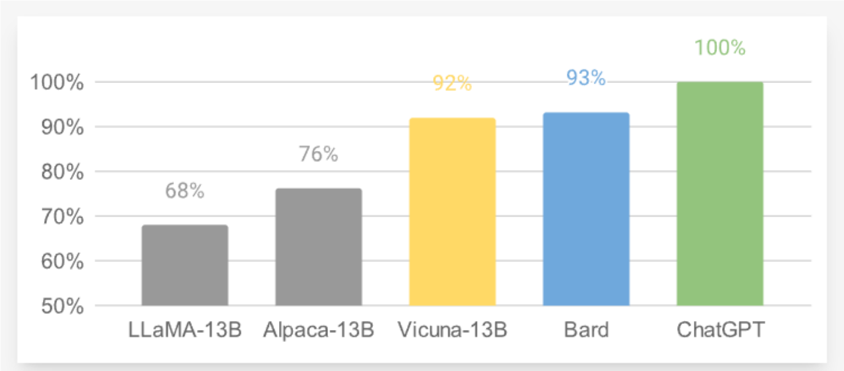


Figure 29 Evaluation of models done by GPT-4
Source: https://lmsys.org/blog/2023-03-30-vicuna/

Power and Capabilities:

- **Compact Size:** Based on the smallest model that has 7 billion or 13 billion parameters, it is very light and easy to set up on small infrastructures.

- **Better performing:** With the addition of over 70k user shared ChatGPT conversations, Vicuna is performing better results.

- **Open-Source:** all the code is available on GitHub.

- **Open for finetuning:** since we have access to the model, we can easily fine tune it to increase is efficiency of response.

- **Total control:** Since the model is hosted on your computer, you can control what you send to the AI, it is important to note that even without internet connection you can still interact with the model.

Limitations

- **Lack of Real-Time Information**: Vicuna knowledge is limited to what it has been trained on which typically includes information available up until the time of its training. It does not have real-time or up-to-date information on current events or recent developments.

- **Storage cost / power consumption**: When deploying the AI model, whether locally or in containers, it is important to consider the associated energy costs. The energy consumption varies depending on the size of the model being used. Additionally storing the models, themselves may require storage and storing cost should also be considered.

- **Small Model:** having a small model has also is disadvantages, the context, coherence and writing performance will not match a big model of over 1 trillion parameters like GPT-4

2.1.3 Gopher

Developed by Deepmind, Gopher is a language model almost as large as GPT-3 with over 280 billion parameters (*Language Modelling at Scale*, n.d.).

Figure 30 Deepmind logo

Source: https://www.enterpriseai.news/2021/12/08/deepmind-experimenting-with-its-nascent-gopher-280-billion-parameter-language-model/

The model was trained on a massive text dataset containing webpages, books, news and more. Based on the reports of the laboratory, Gopher is capable of exceeding some models available on the market for some tasks (Demo, n.d.) (*Google Trains 280 Billion Parameter AI Language Model Gopher*, n.d.), (Schreiner, 2022).

For example, in:

- Humanities
- Social science
- Medicine
- General Knowledge
- Science and Technology
- Maths

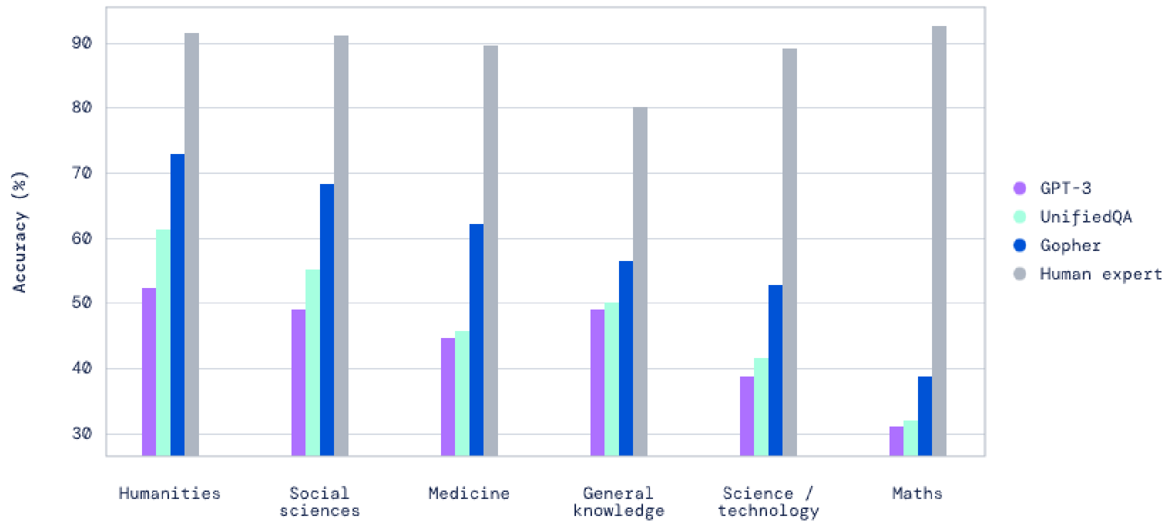


Figure 31 Gopher accuracy evaluation

Source: https://www.deepmind.com/blog/language-modelling-at-scale-gopher-ethical-considerations-and-retrieval

Power and Capabilities:

- **Energy saving:** Since we don't need to deploy a server or tool to manage the AI, the energy cost is inexistent.
- **Model Size:** Gopher has a large model of over 280 billion parameters.
- **Coherence:** This model has a surprising coherence in dialogue interactions
- **Improved performance:** Gopher has great performance in reading comprehension, fact-checking and identification of toxic language and according to the reports exceeds the other models in various tasks.

Limitations

- **Not available for the users**: Despite the good results from this model, it is impossible at the moment to know the limitations and to validate the results of the report of this artificial intelligence without testing it and therefore will not be included in the matrix of comparison.

2.1.4 PaLM 2

Created by Google, Palm 2 is a next generation large language model based on the previous version of PaLM 1 (*Google AI PaLM 2*, n.d.), (*Introducing PaLM 2*, 2023).

Figure 32 Palm 2

Source: https://mashable.com/article/google-io-2023-palm2-ai-announcement

It is effective on:

- Advanced reasoning tasks
- Code problems
- Translations

Palm 2 is used as a base for other large language model as for example Med-Palm, as LLM designed in the medical domain or Sec-Palm, a model designed to protect users from cyber-attacks (*Med-PaLM*, n.d.).

Power and Capabilities:

- **Model Size:** Palm 2 has a large model of over 280 billion parameters.

- **Energy saving:** Since we don't need to deploy a server or tool to manage the AI, the energy cost is inexistent.

- **Nuance understanding:** This model is capable of understanding nuance including riddles and idioms.

- **Reasoning and translation:** Palm 2 achieved great performance in reasoning and translation benchmarks.

- **API:** PaLM 2 has an API that can be used by developers to integrate it into their webapp / tools.
    - According to Google, the API offers fast and efficient performance with speed up to 75+ tokens per second and also offer a context window of 8000 tokens.
    - The API has been designed to respond effectively in:
        - Coding problems

- Writing
- Editing
- Problem Solving
- Recommendations
- Data creation
- Agents

Limitations

- **Not available for the users**: Despite the good results from this model, it is impossible at the moment to know the limitations and to validate the results of the reports of this artificial intelligence without testing the API and the model and therefore will not be included in the matrix of comparison.

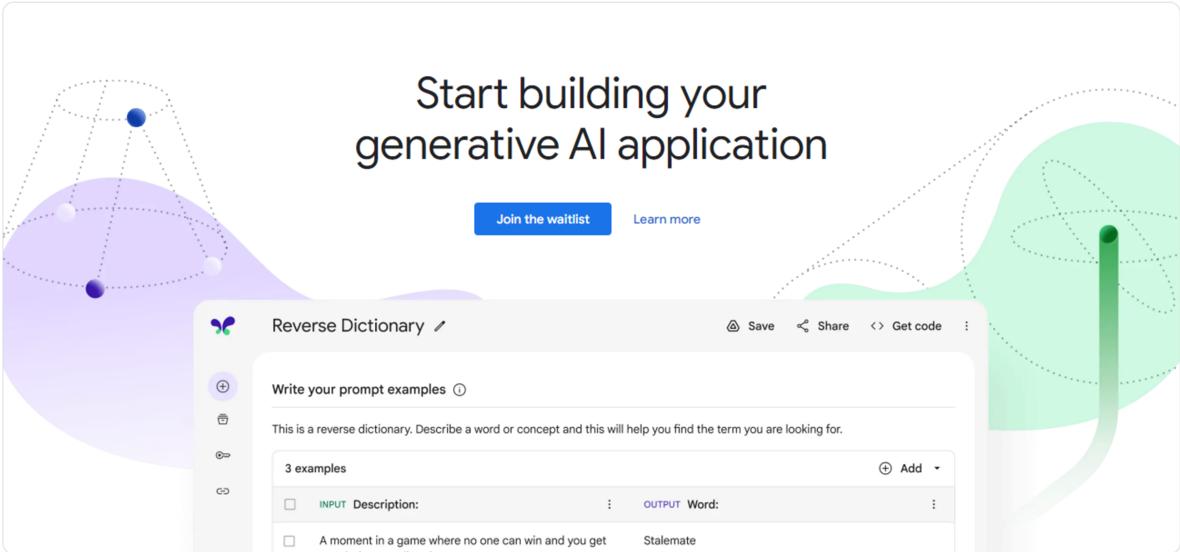


Figure 33 Waiting list

Source: https://developers.generativeai.google/products/palm

2.1.4.1 Bard

Released on March 21 in 2023, Bard was developed by Google and is a direct response to the rise of OpenAI ChatGPT (*Essayez Bard, une expérience d'IA conversationnelle par Google*, n.d.).

Bard is an interface to a large language model (LLM) that gives the ability to users to collaborate with generative AI (*An Important next Step on Our AI Journey*, 2023), (*What Is Google Bard? Here's How to Use This ChatGPT Rival | Digital Trends*, n.d.).

Despite having similarities like the famous chatbot such as ChatGPT, Bard is designed to function as collaborative tool that will allows users to work directly with the generative AI in various contexts such as :

- Creative writing
- Brainstorming
- Problem-solving

Power and Capabilities:

- **Energy saving:** Since we don't need to deploy a server or tool to manage the AI, the energy cost is inexistent.

- **Model Size:** Bard is using a model of over 1.56 billion parameters.

- **Generative AI:** Can be used for various creative purposes, the generative AI is able to create content that is not directly derived from the existing data learned.

- **Access to real-world:** Bard is able to use google search to find information that is up to date, it means that this language model is able to generate text that is more relevant than other closed models.

Limitations

- **Not available for Switzerland:** At the moment Bard isn't available for Switzerland for testing, even the usage of VPN cannot bypass this restriction.

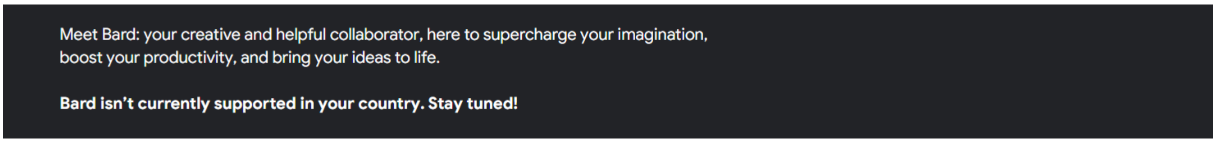


Figure 34 : Bard is not available in Switzerland
Source: https://bard.google.com/

2.2 Analysis of the different libraries & frameworks

2.2.1 Tensorflow

Tensorflow is an open-source library created by Google for fast numerical computing (*TensorFlow API Versions | TensorFlow v2.13.0*, n.d.).

It provides a set of tools and functions that make it easier to work with numbers and create intelligent systems, with it you can represent mathematical operations as a network of interconnected nodes (Inc, n.d.).

The purpose of this library is to facilitate the development and training od machine learning model, you can create neural networks that can learn patterns and do predictions (Nicholas Renotte, 2020), (freeCodeCamp.org, 2020).

2.2.1.1 Key points

1) Flexibility

Tensorflow provides a huge variety of tools,API,modules that support different levels of abstraction.

2) Scalability

By being efficiently scalable, Tensorflow is suitable for both small and very large machine learning projects.

3) Cross platform

Compatibility with Windows / Mac and Linux.

4) High performance

Tensorflow has high optimization techniques that enable efficient executions of computations. This high-performance capability allows for faster model training and inference.

5) Extensive collection of pre-trained models.

TensorFlow offers up to 1300 models used for 4 domains such as computer vision, audio and text.

2.2.2 Pandas

Used for data manipulation and analysis, Pandas provides easy-to use data structures with a range of functions and methods to work with structured data (*Pandas - Python Data Analysis Library*, n.d.).

During my research on how finetuning and training works on models, I often used Pandas to manipulate the dataset, for example I used it to get the dataset into my data variable to be able to pre-process it later (*Pandas Tutorial*, n.d.).

import pandas as pd

dataset_train = pd.read_csv('data\\Google_Stock_Price_Train.csv')7

dataset_total = pd.concat((dataset_train['Open'], dataset_test['Open']), axis = 0)

With Pandas, I was also able to concatenate 2 columns from 2 different dataset easily (*Pandas | Python Library - Mode*, 2016).

2.2.2.1 Key points

1) **Great for data manipulation**

Pandas provide a large panel of functions and methods for manipulating data such as

- Filtering
- Sorting
- Concatenation

It allows us to efficiently perform pre-processing tasks on the datasets.

2) **Implementation of dataframes**

With the introduction to new primary data structures (DataFrames & Series), these structures simplify data handling and enable intuitive operations on structured data.

The dataFrame is a 2-dimensional table with rows and columns and Series are one dimensional labeled array.

2.2.3 PyTorch

PyTorch is an opensource machine learning framework developed by Meta AI Research Lab. The goal of this framework is to provide a flexible and dynamic approach when building deep learning models (*PyTorch*, n.d.), (freeCodeCamp.org, 2022).

Based on Torch, PyTorch is used to train computation vision models like Tesla Autopilot.

2.2.3.1 Key points

1) **Deep learning support**

PyTorch offers a large set of tools and modules for building neural networks including:

- Pre-defined layers
- Activation functions

- Loss functions
- Optimization algorithms

2) Easy to learn

Nicely documented, Pytorch has a big community and non-official tutorials online to help the beginners to understand how to use PyTorch and implement it to their projects.

The tutorial provided by the Official PyTorch team is easy to understand and guides the user by offering a step by step to learn the basics of the framework.

3) Compatibility with other libraries

PyTorch integrates easily with common Python packages and allows users to leverage the power of PyTorch with other tools and frameworks.

4) Dominance on HuggingFace

PyTorch is still at the moment the leading player in the Transformer field of HuggingFace. The vast majority of the models available on it have been trained using PyTorch.

2.2.4 Keras

Keras is an open-source deep learning framework that is designed to be intuitive and modular, it provides a user-friendly and high-level API for building and training neural networks (*Keras: Deep Learning for Humans*, n.d.), (*What Is Keras and Why Is It so Popular in 2023?*, n.d.).

During my research on how to create neural networks I have often used Keras to create Convolutional neural network by importing layers like the one below:

from keras.models import Sequential

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

Keras provided me with pre-made layers that I could use to create image recognition models very easily (freeCodeCamp.org, 2020).

2.2.4.1 Key points

1) Modularity and extensibility

Keras allows users to create neural networks by stacking layers and connecting them together. It provides a range of:

- Pre-defined layers
- Activations functions
- Loss functions

They can be easily combined and customized to create complex models.

2) Documentation & Community

Keras has a large and active community of developers and researchers. The community provides extensive non-official documentation, tutorials and resources to the Keras user. The official website of Keras offers a nice step-by-step tutorial for beginners and provides detailed documentation on how to use this framework.

3) Compatibility

This framework is also designed to work without issue with others popular deep learning framework such as TensorFlow and therefore allowing the user to use the capabilities of Keras with other frameworks.

4) Visualisation & training

Keras also offers tools for visualizing models, inspecting layer activation and monitoring the training process making it easier for the user to know what is happening and debug his neural network.

2.2.5 Accelerate

Accelerate is a python library that provides tools for optimizing and accelerating the performance of deep learning models. The main focus is on hardware acceleration to increase the speed and efficiency of computations (*Accelerate*, n.d.), (*Huggingface/Accelerate*, 2020/2023).

To enable for example GPU acceleration, it is done by a simple setup:

from accelerate import automl

automl.training_loop(device="gpu")

The deep learning process will then take advantage of this GPU acceleration and leads to potentially faster training time.

2.2.5.1 *Key points*

1) Compatibility

Accelerate can be easily integrated into existing deep learning projects seamlessly.

2) Automatic optimization

This tool can automatically apply optimizations improvements to the deep learning models without manual intervention.

3) GPU support

Accelerate provides GPU support and optimizes computations to increase the processing capabilities of the GPU. It gives the ability to models to efficiently use GPU resources and therefore decreases the training time.

2.2.6 Datasets

Datasets is an open-source library developed by Hugging Face. The library provides a wide range of pre-built and ready to use datasets for various machine learning task such as (*Datasets*, n.d.):

- Text classification
- Question answering
- Translation
- Image classification

For example, to import a dataset for your project, you can simply do the following:

```
from datasets import load_dataset
data = load_dataset("DatasetName")
```

Right now, Hugging Face has over 40'000 datasets available for users to download.

2.2.6.1 *Key points*

1) Easy Data Access

Accessing and loading different datasets has been greatly simplified with Datasets and allow the user to switch rapidly between one and other without issue format and structure issue.

2) Large collection available

With over 40'000 datasets available, Datasets offers a vast collection of data that covers a wide range of domains and tasks.

3) Splitting & Shuffling

The dataset library provides functions for splitting the data into train / Validation / test sets. It also offers shuffling to randomize the data samples and avoid biased training.

4) Compatibility

This library is compatible with popular libraries and frameworks and can easily be integrated to existing projects.

2.2.7 PEFT - Parallel Efficient Transformers

Developed by Hugging Face, PEFT is a library that focuses on optimizing the training and inference performance of transformer-based models, especially for language processing tasks (*PEFT*, n.d.).

In the finetuning chapter where we added the LoRA adapters layers to the Alpaca Model to finetuning it, we used the PEFT library to apply this custom optimization technique (Takyar, 2023).

```
from peft import LoraConfig, get_peft_model
config = LoraConfig(
   r=24,
   lora_alpha=16,
   lora_dropout=0.08,
   bias="none",
   task_type="CAUSAL_LM"
)
model = get_peft_model(model, config)
```

1) Training Optimization

PEFT provides efficient parallelization techniques for training transformers models. It optimizes this parallelism within the model and allows for faster training.

2) Memory Optimization

PEFT also provides memory optimization techniques to reduce the memory resources requirement for training by doing gradient checkpointing, a technique used to trade off computational time for memory consumption.

3) Custom Optimization

PEFT gives the ability to the user to specify their own parallelization and optimization strategies by adding for example the LoRA adapters layers.

2.2.8 Transformers

Transformers is a very popular library developed also by Hugging Face, it provides a set of tools for working with transformer-based models in natural language processing. It offers a large range of pre-trained models and fine-tuning capabilities (🤗 *Transformers*, n.d.), (*An Introduction to Using Transformers and Hugging Face*, n.d.).

2.2.8.1 Key points

1) Vast collection of pre-trained models

The library provides a large panel of pre-trained transformer models including BERT, GPT and others. These models have been trained on large-scale datasets and offer a great performance for NLP tasks.

During my research, to save up time and only focusing on finetuning, I was able to start from a pre-trained model using the library and therefore avoid starting from scratch by doing the following:

```
from transformers import AutoModelForCausalLM
model = AutoModelForCausalLM.from_pretrained(
    "NameOfThePreTrainedModel",
    load_in_8bit=True,
    device_map='auto',
)
```

2) Model customization

Transformers provide also a modular design that allows the user to easily configure their models, it provides interfaces for model architectures, tokenizers, and optimizations algorithms.

I also used the tokenizer from the Transformer to acquire the tokenizer of the pre-trained model using:

```
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained("PreTrainedModelName")
```

3) Training & Evaluation

Transformers also contains tools for training transformer models on given datasets, and also for evaluating models based on metrics like accuracy, F1 score and so on.

During my research I used the transformer library for training my models, it was easy to understand and very customizable.

```
trainer = transformers.Trainer(
  model=model,
  train_dataset=data['train'],
  args=transformers.TrainingArguments(
    per_device_train_batch_size=4,
    ...
  ),
  data_collator=transformers.DataCollatorForLanguageModeling(tokenizer, mlm=False)
)
trainer.train()
```

2.2.9 Huggingface_hub

HuggingFace_hub is a library developed by HuggingFace that enables the user to share files to his huggingFace repository (🤗 *Hub Client Library*, n.d.), (*Hugging Face Hub | 🦜🔗 LangChain*, n.d.).

During my research I often used this library to push models to my HuggingFace account for backups like the following:

```
from huggingface_hub import notebook_login
notebook_login()
model.push_to_hub("NameOfTheModel", use_auth_token=True)
```

2.2.9.1 *Key points*

1) File uploading

The library gives the possibility to the user to upload files to the HuggingFace hub without using Git.

2) Flexibility with google co

You can use the login() function to programmatically log in to your Hugging Face account and access the Hub resources. It is done via generating a token from the account page and copy pasting the token in the login popup.

2.2.10 Sklearn

Sklearn is a library used for machine learning, it contains a wide range of tools and functionalities for tasks like data pre-processing, feature selection, training and evaluation (*Scikit-Learn: Machine Learning in Python — Scikit-Learn 1.3.0 Documentation*, n.d.), (*Scikit Learn Tutorial*, n.d.).

One of the strengths of Sklearn is its simplicity and ease of use.

2.2.10.1 *Key points*

1) Wide range of algorithm

Sklearn offers a vast panel of machine learning algorithm, including.

- Classification
- Regression
- Clustering
- Dimensionality reduction

2) Preprocessing tools

The library also offers a collection of tools for manipulating data such as :

- Handling of missing value
- Scaling
- Normalization of data

These techniques of pre-processing are widely due to their effectiveness of transforming raw data into high quality one.

3) Compatibility

Sklearn can easily be integrated with other libraries such Pandas, it will accept without problems the data format provided by them. This allows for efficient data manipulation and transformation while also using Sklean algorithms.

2.3 Analysis of OpenAI integration works in a company / project.

Finetuning OpenAI model to engage in interactive discussions regarding the performances of NBA players.

The project has been done by a content creator named Liam Ottley. His goal was to finetune the OpenAI model to be able to interact it on the performance of NBA players based on a given dataset (Liam Ottley, 2023), (*NBA Players Performance*, n.d.).

2.3.1.1 Data Gathering

The dataset used for this project is available on Kaggle and can be downloaded from it.

Figure 35 kaggle dataset of NBA players performance

Source: https://www.kaggle.com/datasets/thedevastator/unlocking-the-secrets-of-nba-player-performance

In the dataset we can find the performance data of the national Basketball Association players during the 2019-2020 season.

For the example we will only train the model based on the scores of the players.

2.3.1.2 Data preprocessing - cleaning

This step is crucial since we need now to get rid of useless data such as blank rows etc.

Liam did the data pre-processing on excel to save up time but i can briefly show to you in python the equivalent.

The first step he took was to delete all the blank rows of the dataset so the .csv will only contain relevant information for the model. The following can be done by checking if any values is available on a row by looping in it as shown in the code below.

```
def remove_blank_rows(input_file, output_file):
    with open(input_file, 'r') as file:
        reader = csv.reader(file)
        rows = [row for row in reader if any(field.strip() for field in row)]

    with open(output_file, 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerows(rows)
```

Result

GP Games	GS Games	MPG Minu	PPG Point	FGM Field	FGA Field	FG% Field	3FGM Thr	3FGA Thre	3FG% Thr	FTM Free	FTA Free 1	FT% Free	Player	Position	Team
19	19	37.4	33.5	218	432	50.5	46	146	31.5	154	213	72.3	Luka Donc	PG	DAL
20	20	34.7	31.4	214	413	51.8	103	236	43.6	97	107	90.7	Stephen Cı	PG	GS
20	20	35.8	31.1	215	425	50.6	21	62	33.9	170	185	91.9	Shai Gilgec	SG	OKC
16	16	32.6	30.9	182	341	53.4	11	48	22.9	120	200	60	Giannis An	PF	MIL

Figure 36 dataset in details

Source: https://www.kaggle.com/datasets/thedevastator/unlocking-the-secrets-of-nba-player-performance

Fortunately, there isn't more to do, the data is now ready for the next step.

2.3.1.3 Data preprocessing - transformation

As explained in the OpenAI finetuning chapter, there is a specific format that must be respected in order for the model to understand patterns from it (*OpenAI Platform*, n.d.).

The author transformed the dataset into generation pairs using a python script that he generated on ChatGPT by specifying the format needed.

The script takes each row of the dataset and outputs a generation pair that summarize the score of the player defined by the prompt.

```
# Read CSV file
with open('csv_without_Blank_rows.csv', 'r') as f:
    data = list(csv.reader(f))
    headers = data[0]
    data = data[1:]
# Create prompt and completion pairs
pairs = []
for player in data:
    prompt = f"Write a summary of {player[headers.index('Player')]}'s statistics:"
    completion = f"{player[headers.index('Player')]} played {player[headers.index('GP  Games played')]} games, starting {player[headers.index('GS  Games started')]} of them. He had an average of {player[headers.index('MPG  Minutes Per Game')]} minutes per game, scoring {player[headers.index('PPG  Points Per Game')]} points per game. He made {player[headers.index('FGM  Field Goals Made')]} out of {player[headers.index('FGA  Field Goals Attempted')]} field goals, for a field goal percentage of {player[headers.index('FG%  Field Goal Percentage')]}. He made {player[headers.index('3FGM  Three-Point Field Goals Made')]} out of {player[headers.index('3FGA  Three-Point Field Goals Attempted')]} three-point field goals, for a three-point field goal percentage of {player[headers.index('3FG%  Three-Point Field Goal Percentage')]}. He made {player[headers.index('FTM  Free Throws Made')]} out of {player[headers.index('FTA  Free Throws Attempted')]} free throws, for a free throw percentage of {player[headers.index('FT%  Free Throw Percentage')]}. He plays as {player[headers.index('Position')]} for the {player[headers.index('Team')]}."
    pairs.append({"prompt": prompt, "completion": completion})
# Export to JSON file
with open('prompt_completion_pairs.json', 'w') as f:
    json.dump(pairs, f)
```

Result of one generation pairs.

*{"**prompt**": "Write a summary of Luka Doncic's statistics:", "**completion**": "Luka Doncic played 19 games, starting 19 of them. He had an average of 37.4 minutes per game, scoring 33.5 points per game. He made 218 out of 432 field goals, for a field goal percentage of 50.5. He made 46 out of 146 three-point field goals, for a three-point field goal percentage of 31.5. He made 154 out of 213 free throws, for a free throw percentage of 72.3. He plays as PG for the DAL."}*

The dataset is now ready for training the model.

2.3.1.4 Training the model

For the finetuning part, the author used the Curie model and launched the training step with the following command line:

```
(venv) PS C:\Users\Admin\PycharmProjects\gpt-app> openai api fine_tunes.create -t prompt_completion_pairs.jsonl -m curie
Upload progress: 100%|
Upload progress: 100%|
Uploaded file from prompt_completion_pairs.jsonl: file-PyMMU8CFGfvujBNkjVYejSzC
```

Figure 37 creation of the finetuning job

Source: author

After a few minutes the model is now finetuned and ready for inference (*OpenAI Platform*, n.d.).

2.3.1.5 *Testing the finetuned model*

Prompt:

Write a summary of Stephen Curry's statistics

Response

Stephen Curry played 19 games, starting 19 of them. He had an average of 30.4 minutes per game, scoring 32.4 points per game. He made 195 out of 419 field goals, for a field goal percentage of 46.2. He made 38 out of 121 three-point field goals, for a three-point field goal percentage of 31.5. He made 125 out of 158 free throws, for a free throw percentage of 77.8. He plays as PG for the MIA. He plays as PG for the MIA. He plays as PG for the MIA. He plays as PG for the MIA. He plays as PG for the MIA. He plays as PG for the MIA. He plays as PG for the MIA. He plays as PG for the MIA. He plays as SF for the MIA. He plays as SG for the MIA.
ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ

Based on the completion we can see that there are a lot of unintentional repetitions and hallucinations. We will see in the next section what are the key points to extract from this project to better finetune the model.

2.3.1.6 *Key points*

Switching to Davinci

The author suggested switching for Davinci since the model is far more powerful at understanding the context given, based on his words, the Curie model isn't powerful enough to really understand the tasks given by the user. He also advises this change because the training cost is not significantly bigger than the cost for training a Curie model (Greyling, 2022).

Respect the best practices given by OpenAI

It is recommended by OpenAI to apply some best practices to help the AI model respond more

accurately (*OpenAI Platform*, n.d.).
The commons advice given by OpenAI are the following:

1. Each prompt should end with a fixed separator to tell the model where the prompt end
   a. “Write a summary of Luka Doncic's statistics**\n\n###\n\n**”

2. Each completion should start with a whitespace
   a. " Luka Doncic played 19 games”

3. Each completion should end with a fixed stop sequence to notify the model when the completion end.
   a. " Luka Doncic played 19 games**\n**”

4. When interacting with the AI model, always format the prompts with the format used for the training so the model will better know how to manage the sequence.

Hyperparameters modifications

The model from the author has only been trained on 4 epochs, by increasing the number by 8 or 10 we could potentially obtain better results since we allow the model more time to learns the patterns from the generations pairs (*OpenAI Platform*, n.d.)..

The performance of the model can also be increased by playing with the batch_size and the learning_rate_multiplier by keeping in mind that if the training dataset is small the risk of overfitting can potentially occur if we train the model too much (*Fine Tuning*, 2023).

2.4 Analysis of works that link finetuning a model for a company / project

In order to gain a better understanding of the functioning of fine-tuning and its application to the IPESE tools, I conducted also researches to gather information about companies or individuals who have already fine-tuned a model (LLaMA models are often used because they are open source).

Finetuning a Alpaca Model with LoRA for sentiment analysis on tweets about crypto currencies

Authored by Venelin Valkov, the goal of his project was to train an alpaca model of 7 billion parameters on a custom dataset of tweets related to bitcoin sentiment to predict the trend of the crypto market (Venelin Valkov, 2023), (🤗 *PEFT*, 2022/2023).

2.4.1.1 Data Gathering

The dataset that he used is available on Kaggle also and contains tweets from 2018 related to Bitcoin, for each tweets a sentiment value has been defined (*BTC Tweets Sentiment*, n.d.).

The dataset is structured the following way, the column "Tweets" contains the tweets related to bitcoins from 2018, the "date" column is the "date when the tweet has been published and finally the "sentiment" column is the classification of the tweet based on the distinction below:

- 0 = neutral
- 1 = positive
- -1 = negative

Date	Tweet	Sentiment
Fri Mar 23 00:41:34 +0000 2018	Embrace the FUD.	-1

Figure 38 example of the dasaset structure
Source: https://www.kaggle.com/datasets/aisolutions353/btc-tweets-sentiment

2.4.1.2 Data preprocessing – cleaning

According to the author, deleting tweets containing "RT" or URL links could potentially remove a chunck of bad data and therefore improve the quality of the dataset.

Since we the author didn't share the data cleaning, I will provide a short python code on how to do it, so we have the whole overview of the project.

Removing tweets containing "RT" or websites links

```
import csv
import re
input_file = 'tweets.csv'
output_file = 'filtered_tweets.csv'
# Function to check if a tweet contains "RT" or URLs
def is_valid_tweet(tweet):
    if "RT" in tweet:
        return False
    if re.search(r"http\S+", tweet):
        return False
    return True
# Read the input .csv file and filter tweets
with open(input_file, 'r', encoding='utf-8') as file:
    reader = csv.reader(file)
    header = next(reader)
    filtered_data = [row for row in reader if is_valid_tweet(row[1])]
# Write the filtered data to a new .csv file
with open(output_file, 'w', encoding='utf-8', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(header)
    writer.writerows(filtered_data)
```

The cleaning was done by creating a function called "is_valid_tweet" that check if the given tweet contains the word "RT" or contains a web URL and return False if yes.

This function will be used during the reading of the input_file where we will do a loop inside it and check every row of the dataset and include the row if the return of the function is False, meaning it is a valid tweet.

The dataset has been now cleaned and ready for transformation.

2.4.1.3 Data preprocessing - Transformation

To help the model understand the new task to learn, a format of data will be followed. The format will contain the "instruction" that will help the model know what we want from it, "input" which will contains the tweet and finally "output" that tell the sentiment of the tweet (*Stanford Alpaca*, 2023/2023).

Example of generation pairs

```
{
  "instruction": "Detect the sentiment of the tweet.",
  "input": "@p0nd3ea Bitcoin wasn't built to live on exchanges.",
  "output": "Positive"
}
```

To be able to transform the dataset into a JSON containing all the generation pairs, the author used the following script:

```
# Function used for the transformation
def sentiment_score_to_name(score: float):
    if score > 0:
        return "Positive"
    elif score < 0:
        return "Negative"
    return "Neutral"
# Creating the dataset
dataset_data = [
    {
        "instruction": "Detect the sentiment of the tweet.",
        "input": row_dict["tweet"],
        "output": sentiment_score_to_name(row_dict["sentiment"])
    }
    for row_dict in df.to_dict(orient="records")
]
dataset_data[0]
```

The function "sentiment_score_to_name" takes as input the sentiment and convert it into a string to better help the model understand it and force him to respond us in this format. The dataset data is then created, and each row is converted into the desired training format by looping on the cleaned CSV (*Decapoda-Research/Llama-7b-Hf · Hugging Face*, n.d.).

2.4.1.4 Data preprocessing - Tokenization of the Dataset

The author also needed to convert the dataset into a prompt and tokenize it so it can be readable for the model.

```
data = load_dataset("json", data_files="alpaca-bitcoin-sentiment-dataset.json")
data["train"]
```

To transform the generation pairs into tokenized prompts the author used the following scripts:

```
# Function used for the transformation of the generation pairs to a prompt
def generate_prompt(data_point):
  return f"""Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.  # noqa: E501
### Instruction:
{data_point["instruction"]}
### Input:
{data_point["input"]}
### Response:
{data_point["output"]}"""
```

The script takes as input the generation pairs and converts it into a prompt that will be better understood by the model.

Even when converted into a prompt, we still need to convert our data into tokens because the model typically operates on numerical data rather than raw text, to perform this the author used the tokenizer from the pre-trained model.

2.4.1.5 Training - Getting the model

To initiate the training, the author first needed to import the pre-trained model, the model used for the project came from the hugging face platform (*Decapoda-Research/Llama-7b-Hf · Hugging Face*, n.d.).

```
BASE_MODEL = "decapoda-research/llama-7b-hf"
model = LlamaForCausalLM.from_pretrained(
    BASE_MODEL,
    load_in_8bit=True,
    torch_dtype=torch.float16,
    device_map="auto",
)
tokenizer = LlamaTokenizer.from_pretrained(BASE_MODEL)
```

The model is imported using the transformer library and since it comes from a LlaMA model the author used the LlamaForCausalLM class to import it. As we have seen in the finetuning chapter, for better performance during the tokenization the author uses the tokenizer of the pre-trained model instead of creating one (*LLaMA*, n.d.).

2.4.1.6 Training

Everything is now ready for the training. To initiate it the author took the model and applied a quantization to it to decrease resource needs for the training using the PEFT library (*Fine-Tuning LLMs Made Easy with LoRA and Generative AI-Stable Diffusion LoRA | by Xiao Sean | Medium*, n.d.). The config used comes from the LoRA technique, we will not talk deeper about it since a detailed overview of it has been done in the finetuning chapter (*What Is Low-Rank Adaptation (LoRA)? - TechTalks*, n.d.).

```
model = prepare_model_for_int8_training(model)
config = LoraConfig(
    r=LORA_R,
    lora_alpha=LORA_ALPHA,
    target_modules=LORA_TARGET_MODULES,
    lora_dropout=LORA_DROPOUT,
    bias="none",
    task_type="CAUSAL_LM",
)
model = get_peft_model(model, config)
```

The author then instantiates the trainer that will take as inputs the model, the training data, test data, the training arguments and the data_collator (*Data Collator*, n.d.).

```
trainer = transformers.Trainer(
   model=model,
   train_dataset=train_data,
   eval_dataset=val_data,
   args=training_arguments,
   data_collator=data_collator
)
trainer.train()
```

The trainer is then launched using the method “train()” (*Trainer*, n.d.).

2.4.1.7 Evaluation of the finetuned model

The model has been finetuned and ready for interaction, due to the computational limitations I am unable to train the model and perform evaluations on it.

Based on the results from the author the model the accuracy is good but can be more accurate with better data cleaning.

PROMPT	SENTIMENT PREDICTION
Get ready to take short positions	neutral (should have been negative)
$btc-2023-? bull market powered by fud	Positive (true)
Did you know that as soon as $btc broke the sp in the gaussian channel, the bull market started?	Positive (true)

Figure 39 evaluation of the result
Source: https://www.youtube.com/watch?v=4-Q50fmq7Uw

2.4.1.8 Key points

Even without having access to evaluation metrics, according to the results shown in the video the model is performing well but often doesn’t understand the sentiment of the tweet (Venelin Valkov, 2023).

The key points to remember in this project are:

1) **Bad data cleaning from the dataset**

During the data cleaning process, I came across a significant amount of irrelevant information in the dataset that could introduce bias and impact the accuracy of the model and his understanding of the sentiment analysis.

I encountered the following type of data:

- Tags of usernames
- Non-sense sentence that couldn't be classified.
- Ads for other cryptocurrencies

3 Choice of Artificial intelligence

Based on the precedent state of art and the information I gathered about the different artificial intelligence available on the market that could be useful for my bachelor thesis project. I've designed a comparison matrix to provide a broad view of the various AI options accessible to me, which will inform the direction I choose for this project.

3.1 Analysis of the OpenAI and LLaMA models

Additional information : the LLaMA 13B model contains 13billion of parameters and the model used for OpenAI is the GPT-3.5. no finetuning and text-embedding has been used for the test.

I unfortunately couldn't test the LLaMA 65B due to my resource's limitations.

3.1.1 Question 1: What is Energyscope and how it affect switzerland ?

3.1.1.1 OpenAI

As of my last knowledge update in September 2021, there is no information available about Energyscope specifically. It is possible that Energyscope refers to a more recent development or a specific project that has emerged since then. However, I can provide you with some general information about Switzerland's energy landscape and policies.

Switzerland has been actively working towards transitioning its energy sector to a more sustainable and low-carbon future. The country has limited domestic energy resources, with the majority of its energy needs being met through imports. As a result, Switzerland has been focusing on energy efficiency, renewable energy sources, and reducing greenhouse gas emissions.

Switzerland has set several ambitious targets and implemented various measures to achieve its energy goals. Some key initiatives include:

Energy Strategy 2050: This strategy was adopted in 2011 with the aim of reducing energy consumption, increasing energy efficiency, and promoting renewable energy sources. It sets a target of reducing per capita energy consumption by 43% and increasing the share of renewable energy to 50% by 2050.

Energy Efficiency: Switzerland has implemented various energy efficiency measures, including building insulation standards, energy labelling for appliances, and financial incentives to encourage energy-saving measures in households and businesses.

Renewable Energy: The country has been actively promoting the use of renewable energy sources such as hydropower, wind power, solar power, and biomass. Switzerland has set specific targets for increasing the share of renewables in electricity production.

Nuclear Phase-out: Following the Fukushima nuclear disaster in 2011, Switzerland decided to gradually phase out nuclear power. The existing nuclear power plants will be decommissioned at the end of their operational lifetimes, and no new nuclear plants will be built.

Carbon Pricing: Switzerland has implemented a system of carbon pricing to encourage the reduction of greenhouse gas emissions. This includes a domestic emissions trading scheme and a carbon tax on fossil fuels.

These are just a few examples of the initiatives and policies that Switzerland has been implementing to transition its energy sector. However, without specific information about Energyscope, I cannot provide any specific details about its impact on Switzerland.

3.1.1.2 LLaMA 13B

Hello! I'm here to help you with your question. However, I noticed that the question seems to be missing some information. "Energyscope" is not a widely known term, and I couldn't find any information about it. Could you please provide more context or clarify what you mean by "Energyscope" and how it relates to Switzerland?

Additionally, I would like to point out that it's important to be mindful of the language we use and avoid perpetuating harmful stereotypes or biases. Switzerland is a diverse country with multiple languages and cultures, and it's important to respect and appreciate the diversity of its people.

If you have any further questions or clarifications, I'll do my best to provide helpful and accurate information.

3.1.2 Question 2: What are the impact of nuclear energy in Valais Switzerland?

3.1.2.1 OpenAI

Nuclear energy plays a significant role in the canton of Valais, Switzerland. Valais is home to the Bugey Nuclear Power Plant, which is located in the municipality of Bugey in France, near the Swiss border. While the power plant itself is not in Valais, its operations have an impact on the region. Here are some key aspects to consider regarding the impact of nuclear energy in Valais:

Electricity Generation: The Bugey Nuclear Power Plant generates a substantial amount of electricity, which contributes to the overall energy supply in the region. This nuclear power plant, along with other nuclear facilities in neighboring countries, helps meet the electricity demand of Valais and Switzerland as a whole.

Energy Independence: Nuclear power provides a significant contribution to Switzerland's energy independence. By relying on nuclear energy, Switzerland reduces its dependence on imported fossil fuels and diversifies its energy mix.

Economic Considerations: The presence of the Bugey Nuclear Power Plant contributes to the local and regional economy in Valais. It creates job opportunities both directly, through plant operations, and indirectly, through related services and industries. The power plant's operations and maintenance require a skilled workforce, leading to employment opportunities for the local population.

Safety and Environmental Concerns: Nuclear energy production raises safety and environmental concerns. While the Bugey Nuclear Power Plant is subject to stringent safety regulations and oversight, accidents or incidents can have serious consequences. In Valais, there is potential concern regarding the impact of a nuclear accident at the Bugey plant, as it is located relatively close to the Swiss border.

Public Opinion and Nuclear Phase-Out: Public opinion regarding nuclear energy in Switzerland has been mixed. Concerns about safety, waste management, and the risk of accidents have led to debates about the future of nuclear energy. It's important to note that Switzerland has decided to phase out nuclear power gradually, with existing plants scheduled to be decommissioned at the end of their operational lives.

It is crucial to consult up-to-date sources and local authorities for the most accurate and current information on the specific impacts of nuclear energy in Valais, Switzerland.

3.1.2.2 LLaMA 13B

Thank you for your question. I'll do my best to provide a helpful and accurate response.

Nuclear energy is a controversial topic, and its impact can be complex and multifaceted. However, I must point out that Valais, Switzerland does not have any nuclear power plants or any history of nuclear energy production. Switzerland has a strong commitment to renewable energy and has set ambitious targets to reduce its carbon footprint.

Instead, I suggest we focus on the impact of renewable energy sources in Valais, Switzerland. The canton has a rich potential for hydroelectric power, and many of its villages and towns are powered by hydroelectric plants. Additionally, there are also several wind farms and solar parks in the region, which contribute to the local energy mix.

In terms of the environmental impact, renewable energy sources like hydroelectric and wind power have a much lower carbon footprint compared to traditional fossil fuels. They also help to reduce the risk of air and water pollution, which can have positive effects on human health and the local ecosystem.

However, the construction of large-scale renewable energy infrastructure can also have negative impacts on the environment and local communities. For example, the construction of dams for hydroelectric power can disrupt the natural flow of rivers and affect the habitats of local wildlife.

Overall, the impact of renewable energy in Valais, Switzerland is a complex issue, and it's important to consider both the positive and negative aspects when evaluating its effectiveness. I hope this information is helpful. If you have any further questions, please feel free to ask.

3.1.3 Question 3 : Are there any recent updates or developments in Switzerland's energy policies or targets ?

3.1.3.1 OpenAI

As of my last knowledge update in September 2021, I don't have information on the most recent updates or developments in Switzerland's energy policies or targets. However, to get the most accurate and up-to-date information on recent developments in Switzerland's energy sector, I recommend referring to official government sources, such as the Federal Office of Energy (BFE) in Switzerland or relevant Swiss government websites. These sources will provide the latest information on any updates, changes, or new targets related to Switzerland's energy policies.

3.1.3.2 LLaMA 13B

Thank you for your question. I'll do my best to provide a helpful and accurate response.

Yes, Switzerland has recently updated its energy policies and targets to address the challenges of climate change and ensure a sustainable energy future. In 2019, the Swiss government set a new energy strategy that aims to reduce greenhouse gas emissions by 50% by 2030 compared to 1990 levels, and to increase the share of renewable energy in the energy mix to at least 60% by 2035.

To achieve these targets, Switzerland has introduced several policies and measures, such as:

Increasing the use of renewable energy: Switzerland has set a goal to increase the share of renewable energy in the energy mix to at least 60% by 2035. This includes the expansion of hydropower, wind power, and solar energy.

Improving energy efficiency: Switzerland has set a goal to reduce primary energy consumption by 15% by 2030. This includes measures such as building insulation, energy-efficient appliances, and smart grids. Phasing out nuclear power: Switzerland has decided to gradually phase out nuclear power by 2029. The country plans to replace nuclear power with renewable energy sources and increase energy efficiency.

Promoting electric vehicles: Switzerland has set a goal to have 50,000 electric vehicles on the road by 2020. The country has introduced incentives such as tax breaks and subsidies to encourage the adoption of electric vehicles.

Carbon pricing: Switzerland has introduced a carbon pricing system to incentivize companies to reduce their greenhouse gas emissions. The system sets a price on carbon emissions and provides a financial incentive for companies to invest in low-carbon technologies.
These policies and measures are expected to have a positive impact on Switzerland's energy sector and contribute to the country's goal of reducing greenhouse gas emissions. However, it's important to note that the implementation of these policies may have some challenges and trade-offs, and it's important to consider the potential impacts on the economy, jobs, and local communities.

3.2 Evaluation of their score

3.2.1 Evaluation Summary

KEY POINTS	OPENAI	LLAMA 13B
CONTEXTUAL UNDERSTANDING	10	9
ACCURACY	9	6
RELEVANCE	9	8
TOTAL	28	23

Figure 40 evaluation of the answers

Source: author

3.2.2 Contextual understanding

Both AIs show a good understanding of the context but in general OpenAI was able to provide more precise answers. In the question 1, OpenAI was able to provide relevant information even when the term “Energyscope” was unknown. LLaMA au the other hand was less precise.

3.2.3 Accuracy

Both Ais provided accurate informations but OpenAI did provide more accurate answers in general, in the question 2, OpenAI referred to Bugey Nuclear Power Plant and its impact on Valais compared to LLaMA who focused on renewable energy.

3.2.4 Completeness

In the completeness side, we can see that OpenAI seemed more comprehensive in its response, covering a wide range of aspects related to the questions. LlaMA on the other hand deviated a lot in the question 2 by not really answering the question.

3.3 Matrice of comparaison detailed

3.3.1 Capabilities

3.3.1.1 OpenAI

Model size

OpenAi has a lot of capabilities that could be helpful for my bachelor thesis, the biggest model available that can be used with API calls is the GPT-3.5 model and contains 355 billion of parameters (*GPT-3.5 vs GPT-4 - Here’s What We Know so Far?*, 2023).

API

Secondly, One of the capabilities that distinguish openAI from the others is that an API is available and can be used to integrate the model into project, the API can be used on standard models such as the 3.5 model and also the finetuned models (*OpenAI API*, n.d.).

On the other hand, Open-source models such as LLaMA do not have an API since we need to host them locally (*Introduction to Meta AI’s LLaMA*, n.d.).

Finetuning

Finetuning can be done on a panel of models, the choice of the initial model depends on what you want the model to become (classifier, sentiment analysis etc..) (*OpenAI Platform*, n.d.).

Open for Commercial use

OpenAI models can be used for commercial use compared to the Llama Models where they can only be used for research only (*Is It Permissible to Use ChatGPT for Commercial Use?*, 2023).

3.3.1.2 Llama & variants

Model

Leaked on the internet by the community, The LLaMA model size is split into different sizes and therefore can respond to a variety of demands depending on the infrastructure you have (*Introduction to Meta AI's LLaMA*, n.d.).

The model can vary from 7 billion to 65 billion parameters despite the huge gap compared to the OpenAI latest model that contains 1 trillion.

Finetuning

Finetuning Llama & his variants is also possible and can be done using different ways and technique such as PEFT and LoRA (Chris Alexiuk, 2023).

3.3.2 Ease of use

3.3.2.1 OpenAI

Playground available

Testing an OpenAI model is very easy and there are no installations / infrastructures needed to interact with the models. One of the main advantages of the playground is that we can also interact with the finetuned model that we have created (*OpenAI Platform*, n.d.).

Finetuning

The OpenAI model is the easiest to work with, with great documentation available on the OpenAI website, the user can easily follow the step-by-step guide and produce his first finetuned model in a short matter of time. The structure of the JSONL used for the finetuning is easy to understand (*OpenAI Platform*, n.d.).

The whole process can be done on just a command prompt (Windows 11 for me) and just one library ("openai") is needed to perform the finetuning. The time needed to train an OpenAI model is also much shorter than an open-source LlaMA model and doesn't depend on any hardware requirement (CJ Gammon, 2022).

Compared to the training process of Llama models and his variants where must do a lot of steps before starting the training (setting the trainer, the tokenizer, the LoRA adapters etc..) it is much more straightforward and easier to use when going with OpenAI.

The only problems that I personally encountered during the process of finetuning an OpenAI model were as minor as missing brackets in a generation pair compared to the multitude of errors that I had during Llama finetuning.

Sharing

The model can be shared amongst your team by simply sharing the API key (*Possible to Share Finetuned Model with Another User?*, 2021).

3.3.2.2 LLaMA & variants

Learning curve

Learning how the training and finetuning works with open-source models such as LLaMA was very difficult for me, the number of details and steps that need to be understood to train a model is very high and demand a lot of time to be able to output a finetuned model. Compared to the OpenAI ease of use, LlaMA models finetuning should be done with a certain knowledge on deep learning / machine learning to better understand the hyperparameters settings of every component and what are their consequences on the process of finetuning.

Another problem that I often encountered was the lack of examples / tutorials on how to really do finetuning and on a LLaMA model and especially how it works, since the leak is recent there isn't much information about how to play with it at the moment.

Hardware requirement

The hardware requirement needed to finetune a LlaMA model & his variants make it less easy to compare different finetuning to better know which one is the best, I often finetuned OpenAI models with different datasets and with different ways of doing generations pairs and could evaluate them on the OpenAI playground very easily. It wasn't the case with LlaMA models because firstly I didn't have the computational power to train the model and didn't have the financial help to train multitude of models, so it was much more difficult to know if one finetuned model was better than the other one (Pramoditha, 2022), (*What Is the Minimum Hardware Requirement for Trying out Deep Learning?*, n.d.), (*Hardware Recommendations for Machine Learning / AI*, n.d.).

3.3.3 Performance

3.3.3.1 OpenAI

Speed of response

The performance from OpenAI models is the best of the two, the output is fast and coherent. When testing the finetuned models on the OpenAI playground I was astonished to see the speed of response of the models during the evaluations (Vyborov, 2023).

Finetuning process

The finetuning process was also very short since the computational process is done by OpenAI compared to the LLaMA models where you need your own hardware or use third party hardware resources (Caelen, 2023).

3.3.3.2 Llama & variants

Speed of response

The LlaMA performance couldn't have been tested due to my hardware and financial limitation, therefore I had found websites that offered free interactions with LlaMA models and also finetuned models such as Vicuna / Alpaca (*Chat with Open Large Language Models*, n.d.).

The result was satisfying and coherent but was not as fast as the response of OpenAI models. Usually, I had to wait 30 seconds to 45 seconds for a response.

3.3.4 Cost

3.3.4.1 OpenAI

The cost of using OpenAI is the complete opposite to the cost of using LLaMA models.

There is no cost needed in hardware resources and physical infrastructure to use the model and train it. But there are costs when using the API for interacting with the model and also the process of finetuning the model. For more information related to the cost of using OpenAI model, please check the "Finetuning" chapter related to OpenAI (*Pricing*, n.d.).

3.3.4.2 LLaMA & variants

On the other hand, the cost related to using LLaMA models are concentrated in the hardware resources and infrastructure that are needed to host the model and train it (Frąckiewicz, 2023).

The cost needed to interact and train the models are dispatched in those following sections:

- Hosting cost (server)
- GPU / CPU / RAM
- Electricity

There is an underlying expense that should be considered, the training required for individuals to grasp the intricacies of fine-tuning an open-source model, the user must have a decent background in software development / machine learning / deep learning (*Could You Train a ChatGPT-Beating Model for $85,000 and Run It in a Browser?*, n.d.).

There are of course other ways to train and interact with open-source models by using low-cost GPU rental like vast.ai, the prices are low and you only pay the computational resources that has been used (Nordhaus, 2017), (*Rent GPUs | Vast.Ai*, n.d.).

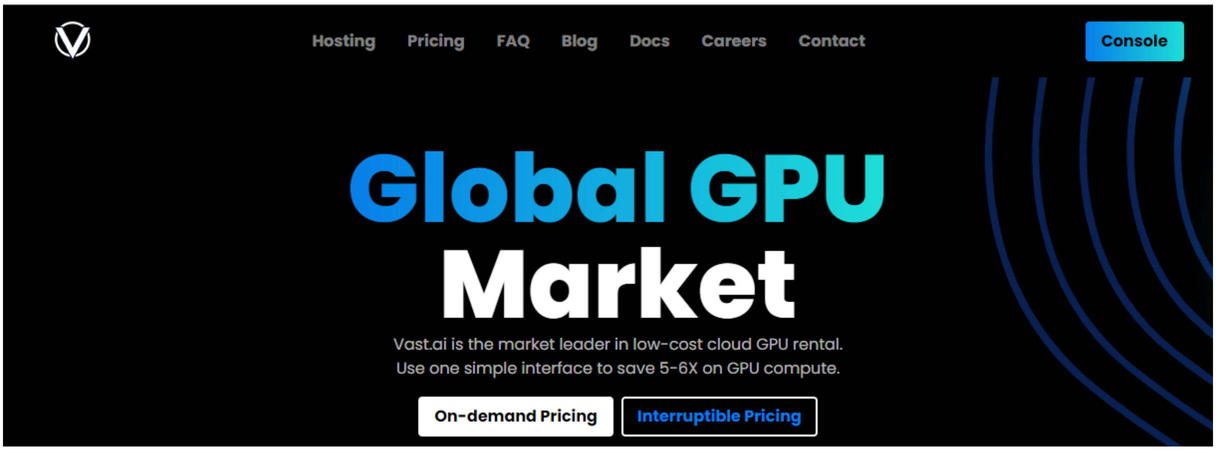


Figure 41 Vast AI website
Source: https://vast.ai/

3.3.4.3 Use case comparison.

Energy scope website wants to integrate a prompt console where users could interact with the AI with those constraints:

- The AI is available 24/24 7/7
- The prompt request is unlimited for users.
    - 1000 users daily
        - 10 questions for each user.

- Each question contains 11 tokens.
  - Example: "What is the capital of Paris? How do you know?”
- The model has been finetuned to answer questions related to Energyscope inputs and results.
  - Finetuned by a dataset of 275’000 tokens (equivalent of a PDF of 200 pages)
    - The finetuning process took 10 hours.

	OPENAI (DAVINCI)	LLAMA (65B 8-BIT VERSION)	LLAMA (65B 8-BIT VERSION) VAST.AI
HARDWARE & INFRASTRUCTURE	None	16’200 CHF	31.2 CHF daily
API CALL	13,2 CHF daily	None	None
FINETUNING	1,65 CHF	None	13 CHF
ELECTRICITY	None	3.36 daily	None

Figure 42 price comparison of the specific use case

Source: author

API calls OpenAI

11 tokens * 10 questions daily * 1000 users daily = 110’000 tokens
110’000 / 1000 = 110 thousands of tokens
110 * 0.1200 CHF = 13.2 CHF (0.12CHF per 1000 tokens) (*OpenAI Platform*, n.d.)

Finetuning cost OpenAI

Based on the price given by OpenAI for finetuning a Davinci model = CHF 0.0300 / 1K tokens
(275’000 / 1000) * 0.03 CHF = 8.25 CHF (*OpenAI Platform*, n.d.)

Hardware & Infrastructure LLaMA

Based on the minimum required advised from the LLaMA community on Reddit (*Hardware Spec for Finetuning >7B Llama · Issue #160 · OptimalScale/LMFlow*, n.d.), (Marie, 2023), (EnPaceRequiescat, 2023).

- 128GB of RAM (Samsung DDR4) = 900 CHF
- GPU: A100 80GB = 10,000 CHF
- CPU: AMD Ryzen Threadripper 3990X 64-Core = 4600 CHF
- Computer rig to handle the components: 700 CHF

Electricity cost for LLaMA

Based on an electricity cost of 0.20 CHF per kWh
Consumption of GPU: 400 W
Consumption of CPU: 280 W
Total consumption of the rig: 700 W
Daily electricity cost: (0.7 kW) * (0.20 CHF/kWh) * (24 hours) = 3.36 CHF

Hardware & Infrastructure LLaMA with Vast.AI

- RAM: Not specified
- CPU: AMD EPYC-Milan
- GPU: A100 PCIE

Price per hour 1,30 CHF

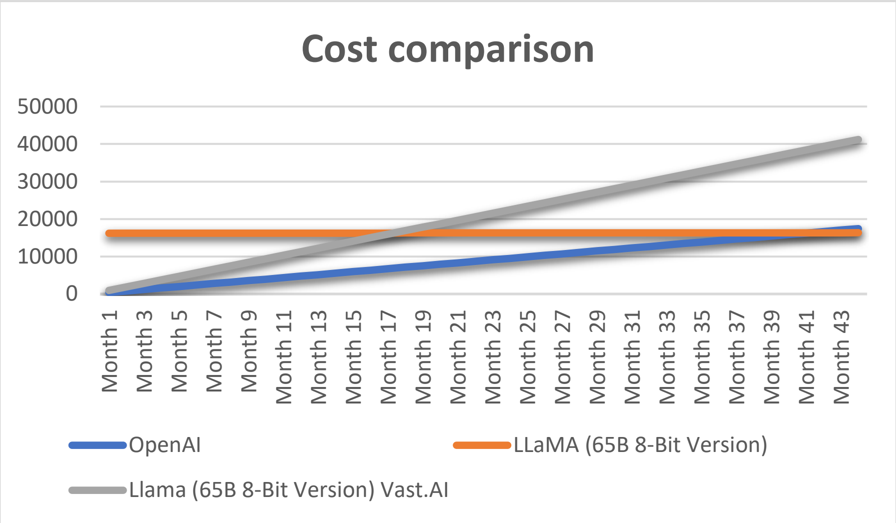


Figure 43 cost comparison of the 3 solutions

Source: author

Based on the graph made with the constraints from the use case, we can draw few conclusions:

- LLaMA (65B 8-Bit Version) is the largest from the start and maintains this position until the vast.ai rental surpasses it at month 18 and by OpenAI at month 42.
- LLaMA (65B 8-Bit Version) is the better solution after reaching 42 months.
- OpenAI is the best solution before reaching 42 months.

3.3.5 Privacy and Security

3.3.5.1 OpenAI

According to OpenAI publication on June 14 on their policies section, starting on March 01 2023, OpenAI will not use anymore the data submitted by users from the API to train and improve their models unless the users specifically tell them to do it. Nevertheless, any data sent through the API will be stored for 30 days for abuse and misuse monitoring (*Safety & Responsibility*, n.d.).

For the security part, OpenAI is often audited by independent third-party and has been granted the SOC 2 Type 2 compliant certification (*Privacy Policy*, n.d.), (*API Data Privacy*, n.d.).

3.3.5.2 LLaMA & variants

Given the model is open-source and completely under your management, you can achieve absolute privacy.

3.3.6 Customizability

3.3.6.1 OpenAI

OpenAi customizability is diverse, for the finetuning part the user can choose between 4 models where each of them has his own capabilities and limitations (*OpenAI Platform*, n.d.).

The finetuning process gives the ability to the user to play with some hyperparameters but not as much when finetuning LLaMA models (*What Is Epoch in Machine Learning?*, 2022), (Zulkifli, 2018), (*Understand Classification Performance Metrics | by Alex Guanga | Becoming Human: Artificial Intelligence Magazine*, n.d.).

The hyper parameters that can be modified are:

- Model type
- Number of epochs
- Batch size
- Learning rate multiplier
- Turning on/off classification metrics

The models can be used for a variety of tasks such as:

- Classification
- Conditional generation

3.3.6.2 LLaMA & variants

The flexibility offered by open-source models like LLaMA surpasses the customization capabilities of OpenAI.

The customization capabilities do not stop at few hyperparameters like OpenAI, it gives the ability for the user to customize:

- The trainer hyper parameters (*Trainer*, n.d.)
- The tokenizer (*Summary of the Tokenizers*, n.d.)
- The whole network
    - Adding LoRA adapters layers (*What Is Low-Rank Adaptation (LoRA)? - TechTalks*, n.d.)
    - Freezing of weights
- Variety of models almost infinite
    - Available on Huggingface.com
- Variety of libraries to use
- Quantization (Nicholls, 2018)

3.3.7 Support & Community

3.3.7.1 OpenAI

The community of OpenAI is very active on the community.openai.com, this website is a developer community forum where developers can ask questions related to OpenAI to the community (*Latest Community Topics*, n.d.).

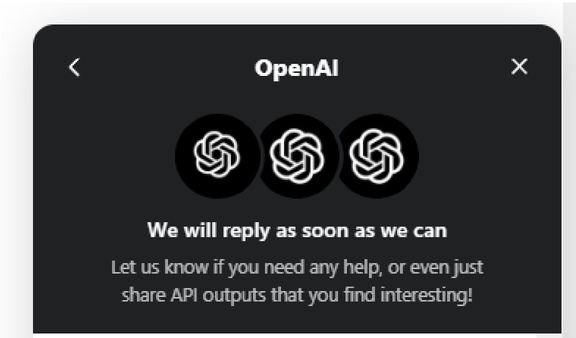


Figure 44 support of OpenAI
Source: Support OpenAI : https://platform.openai.com/docs/guides/fine-tuning/advanced-usage

There are different categories where the user can check and see if his problem has been already resolved. Based on the report of SimilarWeb.com, the forum has a total of visits of 5.9 million people during the last month (*Website Traffic - Check and Analyze Any Website*, n.d.).

The documentation and the guides are also very clear and are available on the official website of OpenAI. Support is also available for users if any problems occur (*OpenAI Platform*, n.d.).

3.3.7.2 LLaMA & variants

On the other hand, despite the vast community behind Open-source models available on Reddit / HuggingFace and so on. There is no official support at the moment for LlaMA models since it was leaked by users on the internet (*LocalLlama*, n.d.).

Nevertheless, there are non-official documentations made by the community that can help the user to better understand the finetuning process (*Hugging Face – The AI Community Building the Future.*, 2023).

3.4 Overview Matrix

CRITERION	OPENAI	LLAMA & VARIANTS
CAPABILITIES	✓	✓
EASE OF USE	✓	✗
PERFORMANCE	✓	✗
COST	✓	✗
PRIVACY & SECURITY	✓	✓
CUSTOMIZABILITY	✓	✗
SUPPORT & COMMUNITY	✓	✓

Figure 45 Matrix of comparison AI
Source: author

3.5 Choice: Open AI API model

I have decided to go with OpenAI API model for the bachelor thesis, OpenAI respond better to the project needs by offering a low-cost API to play with so it will easily integrate within the project, the implementation is fast and does not need hardware resources. The model used for the project will be GPT-3.5.

After assessing the AI comparison, given that we're dealing with a highly specific subject matter, the accuracy of the model is paramount. OpenAI outperformed in this aspect, making it the best fit for our needs.

The ease of use of OpenAI was also an important factor, With the playground I can easily test the model with no cost and therefore evaluate the model very easily. Since I need only one library it will also save me a lot of precious time by avoiding compatibility errors.

Furthermore, the formatting toolkit offered by OpenAI will also save me time by avoiding errors of formatting or converting data into the training format.

By relying on the API only, the performance will be as fast as using the LLaMA model on rental GPU's or by buying the entire hardware and infrastructure at no cost. The cost of the API call is low and based on the graph shown for the use case example it will approximately be the best price option for at least 42 months.

The privacy & security is also an important factor and by being audited by third party and granted the SOC 2 Type 2 compliant certification, The data given trough the API will only be held for 30 days for abuse and monitoring before being deleted, it is for me secure and private enough for the project despite being less private than using an open-source model such as LLaMA.

The customizability provided by OpenAI is diverse enough to potentially modify the effectiveness of the finetuning by playing with the epochs, batch sizes and the rate multiplier and the AI model calls parameters.

The documentation offered by OpenAI is very detailed and the simplicity of the guides for setting up a finetuned or an embedding model will make the process significantly easier for me.

4 Choice of framework that link the AI with the vector database

Using the right library / framework to link the vector database and the AI model is very important, the bachelor project need a library that is easy to integrate, scalable for incoming queries or potential project growth, performant and low cost. Here is a detailed overview on the difference between Haystack and LangChain.

4.1 Description of LangChain

Founded by Harrison Chase, LangChain is an open-source framework designed for building applications that leverage language models. It allows the creation of data-aware applications that can interact with various data sources and perform tasks in their environment.

LangChain works with components and chains:

- Components
  - These are modular and easy-to-use abstractions for working with language models. The components offer several implementations that can be used independently or in conjunction with the entire LangChain framework (*Components | 🦜🔗 LangChain*, n.d.).

- Chains:
  - Structured assemblies of components aimed at accomplishing specific high-level tasks (*Chains | 🦜️🔗 LangChain*, n.d.).

LangChain offers standard, extendable interfaces and integrations for several modules including Model I/O, Data Connection, Chains, Agents, Memory, and Callbacks (*Modules | 🦜️🔗 LangChain*, n.d.).

4.2 Description of Haystack

Founded by Deepset, Haystack is an open-source framework for building production-ready applications using the latest language models. It provides comprehensive tooling, including preprocessing, pipelines, and evaluation (*What Is Haystack?*, n.d.).

Haystack enables semantic search and question answering with advanced transformer-based language models like GPT-3.5. Like LangChain, it also allows flexible use of any Transformer model from Hugging Face's Model Hub. The framework supports end-to-end product development with numerous resources such as file converters, models, labeling tools, and a REST API (*Pipeline Components Overview*, n.d.), (*Language Models*, n.d.).

4.3 Matrix of comparison detailed

4.3.1 Capabilities

4.3.1.1 LangChain

LangChain can be used for a lot of uses cases, it is a rich framework containings a variety of modules that can achieve (*Modules | 🦜️🔗 LangChain*, n.d.):

- Agent simulations
- Agents
- Interaction with APIs
- Autonomous agents
- Chatbots
- Code understanding
- Extraction
- Multi-modal
- QA and chat over documents
- Summarization
- Analysis of structured data

4.3.1.2 Haystack

On the contrary, Haystack framework isn't as rich in features as LangChain, here is the use cases possible with it (*Haystack Introduction*, n.d.):

- Semantic search System
- Information Extractor
- QA question answering

4.3.2 Ease of integration

4.3.2.1 LangChain

LangChain offers a highly convenient and straightforward integration process. The comprehensive documentation provided on LangChain's official website showcases its compatibility with over 37 different vector stores (*Vector Stores | 🦜🔗 LangChain*, n.d.). These include popular and diverse options like:

- Alibaba Cloud
- OpenSearch
- AnalyticDB
- Annoy
- Atlas
- AwaDB
- Azure Cognitive Search
- Cassandra
- Chroma
- Clarifai
- ClickHouse Vector Search
- Activeloop's Deep Lake
- DocArrayHnswSearch
- DocArrayInMemorySearch
- ElasticSearch
- Etc..

Furthermore, the ease of integrating LangChain extends to its programming interface as well. The architecture of LangChain classes and methods has been designed to simplify the connection between different models, reducing the technical overhead for developers (*QA and Chat over Documents | 🦜🔗 LangChain*, n.d.).

Instantiating OpenAI AI model

```
llm = ChatOpenAI(
    openai_api_key=OPENAI_API_KEY,
    model_name='gpt-3.5-turbo-16k',
    temperature=0.0
)
```

Instantiating OpenAI embedding model

```
embed = OpenAIEmbeddings(
    model=model_name,
    openai_api_key=OPENAI_API_KEY
)
```

Connecting the AI model to Pinecone and the embedding model

```
qa = RetrievalQA.from_chain_type(
   llm=llm,
   chain_type="stuff",
   retriever=vectorstore.as_retriever()
)
```

This minimalistic, intuitive coding approach contributes to LangChain's overall ease of use. As such, LangChain is a user-friendly framework that balances powerful functionality with seamless integration and usability (*Integrations | 🦜🔗 LangChain*, n.d.).

4.3.2.2 Haystack

While Haystack provides an accessible and user-friendly integration process, it doesn't quite match LangChain's versatility when it comes to compatibility with vector stores. Haystack's documentation indicates compatibility with a total of nine vector stores (*Retriever*, n.d.):

- Elasticsearch
- In Memory
- Milvus
- OpenSearch
- Pinecone
- Qdrant
- SQL
- Weaviate

- FAISS

On the other hand, LangChain significantly expands on this, showcasing compatibility with over 37 diverse vector stores. This extensive range provides more options and flexibility for developers, accommodating a wide variety of use cases and system environments.

However, it's important to note that despite this, Haystack still maintains a reputation for its ease of integration. It operates using reader and retriever models, and its framework can be integrated swiftly and efficiently, often with just a few lines of code.

While it may offer fewer options for vector store compatibility compared to LangChain, Haystack's user-friendly interface and simple integration process still make it an appealing choice for developers.

10 lines of code needed for a QA functionality

```
document_store = InMemoryDocumentStore()
dicts = convert_files_to_dicts(doc_dir)
document_store.write_documents(dicts)
retriever = BM25Retriever(document_store)
model_name = "loki23/BLOOM-13B"
reader = FARMReader(model_name)
pipe = ExtractiveQAPipeline(reader, retriever)
question = "What is Energyscope ?"
prediction = pipe.run(query=question)
print_answers(prediction)
```

The document store is where the instantiation of the database, in this example the database will be in memory, meaning its contents will be erased once the program is terminated (*Getting Started with Haystack*, n.d.).

The dicts variable contains the conversion of files into documents, it will be used to load the documents into the document store using .write_documents(dicts) method (*DocumentStore*, n.d.).

The retriever is the component that is responsible for selecting the most relevant documents from a large document corpus in response to a user query. The BM25 is a type of retriever based on the “Best Matching 25” algorithm (*Retriever*, n.d.).

The reader is the component used for reading and understanding the text, the retriever gets the text and the reader reads it (*Reader*, n.d.). It uses the “BLOOM” model which has 13 billion parameters (*Bigscience/Bloom-7b1 · Hugging Face*, n.d.).

The pipe will connect the reader and the retriever, it will be used with the .run(query) where the user will give his question.

4.3.3 Scalability

4.3.3.1 LangChain

LangChain achieves scalability trough integration with various vector stores designed for large-scale vector data and by allowing the usage of advanced language models such as GPT-3.5 (*Deployment | 🦜🔗 LangChain*, n.d.).

This dual approach ensures that LangChain can effectively manage and scale with big and complex datasets and provide high-quality responses even if the complexity of the tasks increases.

4.3.3.2 Haystack

Haystack let you build semantic search and question answering applications that can scale up to millions of documents easily (Rehberg, 2022).

The flexible design of Haystack, similar to LangChain, allows you to interchange various AI models as needed at any given time.

4.3.4 Performance

4.3.4.1 LangChain

LangChain performance comes from using highly efficient vector stores and advanced LLM such as GPT-3.5. The performance will depends on the availability of models that are called with APIs or by the resources of the server where your open-source model is running (Politi, 2023).

4.3.4.2 Haystack

Same as LangChain, the performance is dependent on which AI model you choose to work with and which vector database you will use to store the embedded text (*Haystack Benchmarks*, n.d.).

Both of them have the potential for high performance, provided the underlying technologies are chosen wisely.

4.3.5 Community and Support

4.3.5.1 LangChain

Both LangChain and Haystack are well-documented, offering an abundance of guides and quick tutorials to simplify the learning process for users. A multitude of YouTube videos provide further learning support for LangChain. When it comes to community engagement and size, LangChain outshines Haystack, benefitting from a more extensive and active user base (zchaarm, 2023).

Figure 46 langchain github
Source: https://github.com/langchain-ai/langchain

There is no official support unfortunately for LangChain since it is open-source and free (Chase, 2022/2022).

4.3.5.2 Haystack

Figure 47 haystack github
Source: https://github.com/deepset-ai/haystack

Despite being a robust framework, Haystack does not match the level of community activity and engagement seen with LangChain. By being an open-source and free framework, it does not offer official support (Pietsch et al., 2019/2019).

4.3.6 Cost

4.3.6.1 LangChain

LangChain is an open-source framework and free to use.

4.3.6.2 Haystack

Just like LangChain, Haystack is an open-source framework and free to use.

4.4 Overview matrix

MODEL	CAPABILITIES	EASE OF INTEGRATION	SCALABILITY	PERFORMANCE	COMMUNITY AND SUPPORT	COST
LANGCHAIN	✓	✓	✓	✓	✓	✓
HAYSTACK	✗	✓	✓	✓	✗	✓

Figure 48 evaluation of the two frameworks

Source: author

4.5 Choice: LangChain

For the bachelor project, my choice went with LangChain. This comprehensive framework offers a wide array of tools suitable for numerous use cases, setting it apart from Haystack.

Despite the ease of integration being comparable for both Haystack and LangChain, the components of LangChain resonated more with me personally.

When it comes to scalability and performance, LangChain is as good as Haystack. In terms of cost, both are open-source and free, which is a huge plus for both.

One of the determining factors for my choice was the community. LangChain has a significantly larger community than Haystack. This is a crucial aspect for me, as I prefer working with a regularly updated framework, backed by an active community.

5 Choice of vector database

By using embedding we need to use a vector database that could handle the embedded text so our artificial intelligence can better know the context of the question and gain additional knowledge. The comparison will be done on 8 keys points to prove my choice.

5.1 Description of Pinecone

Pinecone is a closed-source vector database for building high-performance AI applications. It supports the latest AI models and users can start building AI-powered applications easily using the Pinecone API (*Vector Database for Vector Search | Pinecone*, n.d.).

5.2 Description of ChromaDB

Chroma is an open-source database for embedding storage, document and query embedding, and embedding search. Its focus is on simplicity, developer productivity, and analysis capability (*Chroma*, n.d.).

5.3 Matrix of comparison detailed

5.3.1 Ease of Use

5.3.1.1 Pinecone

Pinecone shines with its comprehensive and user-friendly dashboard. Accessible at https://app.pinecone.io/, the dashboard allows you to easily create and manage indexes and collections. You can also handle API keys and set member permissions directly from the dashboard, simplifying the management of security settings (*Product | Pinecone*, n.d.).

The dashboard offers an overview of each index, making it easy to monitor the performance and status of each vector database. This high level of visibility can help you identify and resolve issues quickly, as well as understand the usage patterns of your databases.

This ease of use and powerful management capabilities make Pinecone an attractive choice for teams who want to minimize administrative overhead and focus more on their core development tasks. By being automatically pushed on the cloud (Google cloud platform) the dev team doesn't need to worry about further implementations tasks (*Pinecone Is Now Available on the Google Cloud Marketplace | Pinecone*, n.d.).

The documentation & guides are very detailed and provides useful information on how to implement Pinecone in our project (*Overview*, n.d.).

The operations on the DB can either be done via code or directly on the website under the section "Operation".

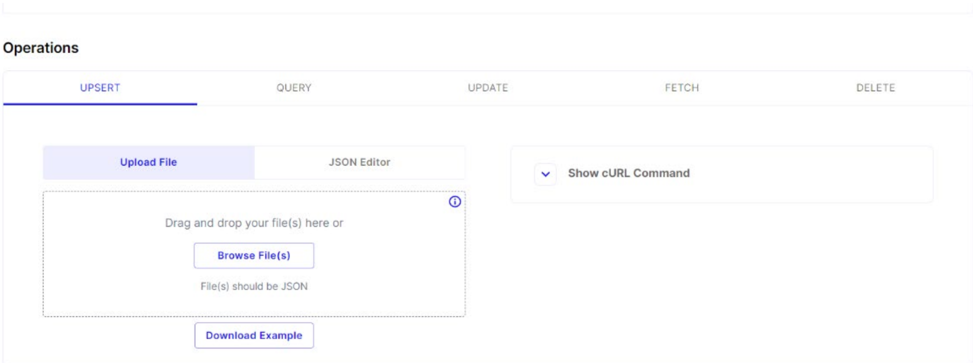


Figure 49 pinecone dashboard
Source: https://www.pinecone.io/

Finally, the introduction of “Collection” greatly simplifies the management of backups and therefore avoid data loss if the database has been corrupted.

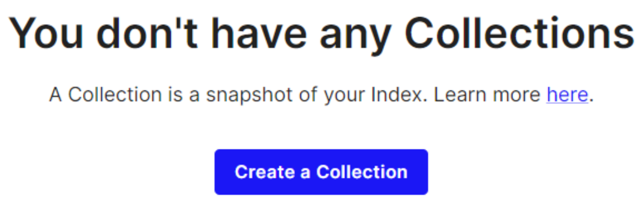


Figure 50 no collections created on pinecone
Source: https://www.pinecone.io/

5.3.1.2 ChromaDB

Given that it is open source, ChromaDB is difficult to manage:

- No graphical user interface to manage all your databases
- No ready to use monitoring available

Everything in ChromaDB must be done via code and therefore make it greatly difficult to manage. Since the ChromaDB is local there is also a further step to do make it available via an API by deploying it into a cloud platform (the creators advise for EC2 on AWS) (🧪 *Usage Guide | Chroma*, n.d.).

The documentation is ChromaDB is acceptable but was not very clear to me for his implementation in my project, I often had to check community forums and YouTube videos to know what I had to do.

5.3.2 Scalability

5.3.2.1 Pinecone

Since it is hosted on the cloud, the scalablity of the vector database are not impacted by our machines, Pinecone offers 3 types of “Pods” (pre-configured units of hardware for running Pinecone services)

POD TYPE	DIMENSIONS	ESTIMATED MAX VECTORS PER PODS
P1	512	1’125’000
	68	1,000,000
	1024	675,000
P2	512	1,250,000
	768	1,100,000

	1024	1,000,000
S1	512	8,000,000
	768	5,000,000
	1024	4,000,000

Figure 51 pod comparison

Source: https//www.pinecone.io/

By having such space, the vector databases on Pinecone are high scalable for unexpected workload size .

5.3.2.2 ChromaDB

Due to the local hardware requirement, ChromaDB scalability depends on the user infrastructure or the user cloud plan (*Vector Databases as Memory for Your AI Agents | by Ivan Campos | Sopmac AI | Medium*, n.d.).

5.3.3 Performance

5.3.3.1 Pinecone

Pinecone performance is very satisfying, by allowing a maximum size of upsert of 2MB, the basic manipulation such as inserting, deleting, fetching is very fast. According to the benchmarks test done by Pinecone, the search speeds of P1 (pod type) is well below 120ms from 0 to 10 millions of vectors on 768 dimensional vectors (*Testing P2 Pods, Vertical Scaling, and Collections | Pinecone*, n.d.).

Pinecone also provides tips & tricks for getting the best performance from their vector databases by:

- Deploying the Pinecone service on the same region
- Reuse connections by using the same pinecone instance

Here is the overview of the performance of the different pods before and after the February 16 update 2022 (*Vector Search Just Got up to 10x Faster, Easier to Set up, and Vertically Scalable | Pinecone*, n.d.).

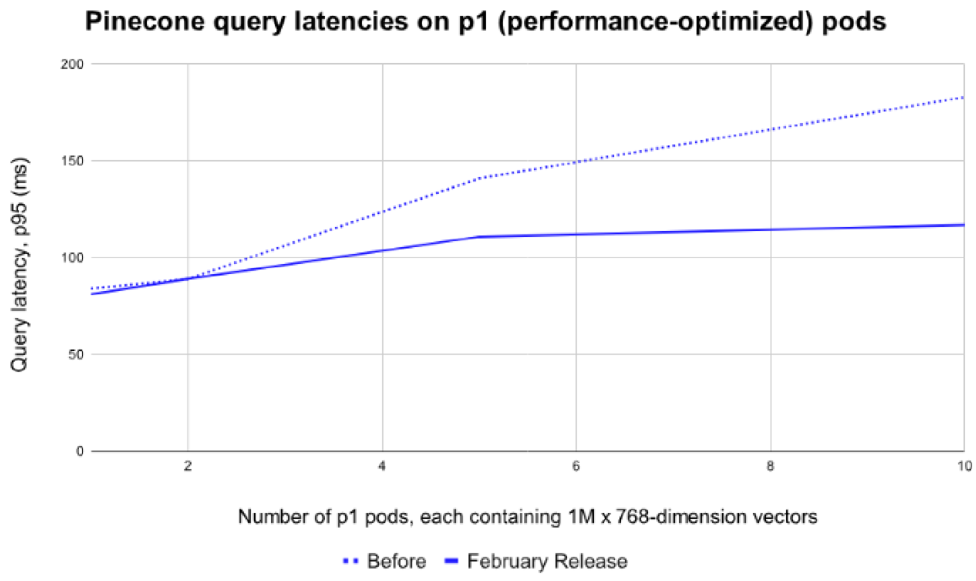


Figure 52 p1 pod evaluation

Source: https://www.pinecone.io/learn/testing-p2-collections-scaling/

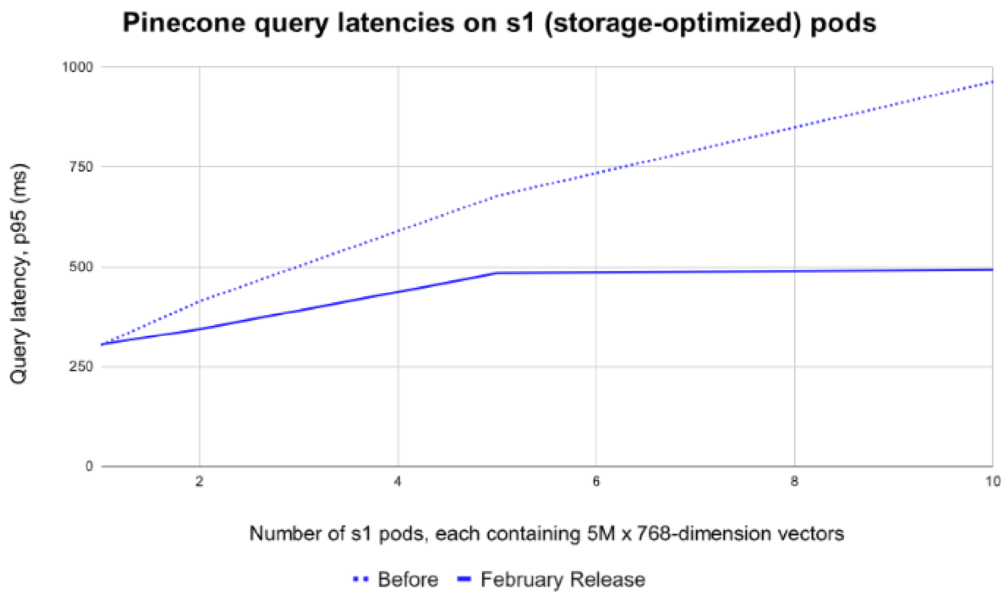


Figure 53 s1 pod evaluation

Source: https://www.pinecone.io/learn/testing-p2-collections-scaling/

5.3.3.2 ChromaDB

Written in Go, ChromaDB is also popular for its high performance and efficiency, during the testing of this library I was also surprised to see the speed of this vector database.

5.3.4 Cost

5.3.4.1 Pinecone

Despite not being open source, Pinecone offers a variety of plans to satisfy the customers needs, for the bachelor project the Starter plan was sufficient since I do not need multiples pods (*Pricing | Pinecone*, n.d.):

- Starter
  - Free
  - Single pod
  - Shared environment
  - No support
- Standard
  - Paid: 70 CHF monthly
  - Unlimited pods
  - Zero-downtime scaling
  - Collection available
  - Choice of cloud region
  - Multiples projects management and users
  - Standard support
    - Email support
    - Technical contacts
- Entreprise
  - 104 CHF monthly
  - As Standard plan
  - Prometheus metrics
  - 24/24 7/7 support
  - Multiple payment options

5.3.4.2 ChromaDB

ChromaDB is an opensource library and therefore is completely free.

5.3.5 Storage

5.3.5.1 Pinecone

The data stored on pinecone databases are either stored on google cloud plateform or on Amazon cloud. The choice is available when the user chooses to go with the standard or entreprise plan (*Inside the Pinecone | Pinecone*, n.d.), (*Product | Pinecone*, n.d.).

5.3.5.2 ChromaDB

The data stored on ChromaDB is completely under control by the user, it can be stored either locally or on a cloud platform chosen by the user.

5.3.6 Security

5.3.6.1 Pinecone

Since it is not open source, it is very important that Pinecone can offer a great security and data encryption for my bachelor project. Pinecone is GDPR-ready and SOC2 Type II certified. The data is encrypted at rest and in transit (*Security | Pinecone*, n.d.).

5.3.6.2 ChromaDB

Despite extensive research into data security measures of ChromaDB, I was unable to locate any relevant information. Consequently, it seems ChromaDB does not inherently incorporate security handling, implying that users may have to implement their own security measures.

5.3.7 Integration

5.3.7.1 Pinecone

The integration of Pinecone is extremely easy, the only library needed is pinecone-client.
The initialisation of pinecone is done by calling the Pinecone.initby also specifying the API key and environment of your database (*Quickstart*, n.d.).

pinecone.init(api_key="YOUR_API_KEY", environment="YOUR_ENVIRONMENT")

After the initialisation, you can simply create an index or connect to an existing one.

Connect

index = pinecone.Index("quickstart")

Create

pinecone.create_index("quickstart", dimension=8, metric="euclidean")

And then insert the desired data

```
index.upsert([
    ("A", [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]),
    ("B", [0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2]),
    ("C", [0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3]),
    ("D", [0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4]),
    ("E", [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5])
])
```

You can also get easy statistics when calling the following method to get an overview of your index.

```
index.describe_index_stats()
```

Pinecone can easily be integrated with OpenAI and others LLM such as LLaMA & hugging face models

5.3.7.2 ChromaDB

ChromaDB integration is done by first installing the library “chromadb”. Since we are using an open source alternative we also need also libraries such as torch, sentences-transformers etc. To grasp all the functionalities of chromaDB. This can potentially lead to bugs and compatibility issues as multiples libraries are concerned (🧪 *Usage Guide | Chroma*, n.d.), (Pradip Nichite, 2023).

The integration is done by initializing the chroma client

```
chroma_client = chromadb.Client()
```

After you can create collections (where you store your embeddings & metadata)

Creating collection

```
collection = chroma_client.create_collection(name="my_collection")
```

Add documents to the collection

```
collection.add(
    documents=["This is a document", "This is another document"],
    metadatas=[{"source": "my_source"}, {"source": "my_source"}],
    ids=["id1", "id2"]
)
```

The database can also be persisted and retrieved for next sessions *client = chromadb.Client(Settings(*

chroma_db_impl="duckdb+parquet",

persist_directory="/path/to/persist/directory" # Optional, defaults to .chromadb/ in the current directory

))

ChromaDB can easily be integrated with OpenAI and others LLM such as LLaMA & hugging face models like Pinecone.

5.3.8 Support and Community

5.3.8.1 Pinecone

Pinecone offers a support and technical support for those for go for paid plans (Standard & enterprise), The is also a community forum hosted by Pinecone that helps the users write their questions and see if his questions has already been posted by someone else (*Community | Pinecone*, n.d.).

There is also a vast number of tutorials on youtube since Pinecone is one of the top picked vector database option when using the LangChain library (Pradip Nichite, 2023).

5.3.8.2 ChromaDB

ChromaDB offers no support, there is a discord community server where users can chat about their problems with other users. Like Pinecone, there is also a vast amount of tutorial on Youtube for getting started with Pinecone (Sam Witteveen, 2023), (*Discord - ChromaDB*, 2023).

5.4 Overview Matrix

CRITERION	PINECONE	CHROMADB
EASE OF USE	✓	✗
SCALABILITY	✓	✗
PERFORMANCE	✓	✗
COST	✓	✓
STORAGE	✓	✗
SECURITY	✓	✗
INTEGRATION	✓	✓

SUPPORT/COMMUNITY	✓	✓

Figure 54 Evaluation between Pinecone and ChromaDB

Source: author

5.5 Choice: Pinecone

For the bachelor project I decided to go with the Pinecone database solution. It is a very easy to use database, with the GUI interface I can easily manage the different indexes of my project and monitor them.

The high scalability of Pinecone makes it also a greater choice since I can add up to 4 millions vectors and still have a very fast database. The performance is great and the respond time is usually very fast.

The cost of using Pinecone is free for my case and therefore is at the same rank as ChromaDB for this point.

Despite the data not being stored in Switzerland, Pinecone meets high security certifications and encrypt rest data and the one in transit making it a secure choice.

The integration part is a personal, when testing both options I was less difficult for me to integrate Pinecone to the existing project than ChromaDB.

Even if the support is not available for starter plans, if one day the bachelor project go into production I can easily switch to a plan with technical and email support, Like ChromaDB, the community and tutorials on YouTube are vast for Pinecone and therefore accentuate my choice for this technology.

6 Choice of embedding model

Choosing the right model of embedding is crucial for the bachelor project, the embedding model must be fast, performant, low cost and create high quality embeddings so the AI model can output text can his coherent and accurate.

6.1 Description of the OpenAI embedding model

The "text-embedding-ada-002" model, released by OpenAI on December 15, 2022, stands as the most recent addition to their suite of embedding models. It surpasses its predecessors in terms of power and cost-efficiency (*New and Improved Embedding Model*, n.d.).

6.2 Description of the Open-source model

The open-source model used is named “sangmini/msmarco-cotmae-MiniLM-L12_en-ko-ja” and is available on Huggingface.com (*Sangmini/Msmarco-Cotmae-MiniLM-L12_en-Ko-Ja · Hugging Face*, n.d.).

It is a sentence-transformers embedding model capable of embedding text at 1536 dimensions just like the "text-embedding-ada-002" model (*Pretrained Models — Sentence-Transformers Documentation*, n.d.).

6.3 Analysis of the OpenAI and open-source embedding models

6.3.1 Elapsed time for embedding and pushing the dataset to Pinecone

The dataset contains all the knowledge about Energyscope and energy in general, amassing a total of 26 documents that collectively consist of 448,646 words.

The model used for generating the output will always be OpenAI, we are here comparing the embedding model and not the LLM model.

6.3.1.1 Using OpenAI embedding model (38 seconds)

```
100%|██████████| 131/131 [00:38<00:00,  3.39it/s]
```

Figure 55 time elapsed for embedding the dataset using OpenAI
Source : author

6.3.1.2 Using open-source embedding model (1 minute and 23 seconds)

```
100%|██████████| 131/131 [01:23<00:00,  1.57it/s]
```

Figure 56 time elapsed for embedding the dataset using open-source embedding model
Source: author

6.3.2 Question 1 : What is the role of Valais in the energy transition ?

QA version : mean we use the RetrievalQA.from_chain_type method of LangChain
QA version with source : mean we use RetrievalQAWithSourcesChain.from_chain_type method of LangChain

6.3.2.1 Using OpenAI embedding model and his embedded data QA version

Based on the provided context, it is mentioned that Valais is a canton in Switzerland. The context also mentions the "energetic strategy" of Valais and the comparison of different energy models for the canton. However, there is no specific information about the role of Valais in the energy transition. Therefore, it is not possible to determine the exact role of Valais in the energy transition based on the given context.

6.3.2.2 Using OpenAI embedding model and his embedded data QA version with source

The role of Valais in the energy transition is to implement an energy independent and CO2 neutral energy system. This involves designing and deploying renewable energy technologies in symbiosis with existing grids and infrastructure. The energy system model used for this analysis is called EnergyScope, which characterizes different energy vector grids and storage. The methodology considers economic and technical parameters to integrate these grids into the overall energy system. The case study focuses on a country with a high existing transmission infrastructure density, which suggests that Valais plays a significant role in the energy transition.
/// SOURCES //https://infoscience.epfl.ch/record/282912?ln=en, https://www.frontiersin.org/articles/10.3389/fenrg.2023.1164813/full

6.3.2.3 Using open-source embedding model and his embedded data QA version

Based on the provided context, it is mentioned that the canton of Valais in Switzerland has developed an energetic strategy called "energetic strategy Valais" with the aim of achieving energetic independence until 2060. The strategy focuses on reducing energy consumption, improving energy efficiency, and meeting residual energy needs with locally produced renewable energy. It also emphasizes the importance of having renewable energy production infrastructure, transmission and distribution networks, and energy storage units mainly in Valaisan hands.

6.3.2.4 Using open-source embedding model and his embedded data QA version with source

The role of Valais in the energy transition is to achieve energetic independence until 2060 by reducing energy consumption, improving energy efficiency, and meeting residual energy needs with locally produced renewable energy. The aim is for renewable energy production infrastructure, transmission and distribution networks, and energy storage units to be mainly in Valaisan hands. The energetic strategy of the canton of Valais is validated through modeling and optimization on a larger scale, considering Valais as part of Switzerland. The Energyscope model is used to determine the impact of various indicators on different energy scenarios and to develop new scenarios. The model has been

modified to generate a regionalized model for Valais, taking into account the potential and demand estimation for the different regions.
/// SOURCES /// https://infoscience.epfl.ch/record/282912?ln=en

6.3.3 Question 2 : What is the parameter Energy scale price and what is value will impact ? provide also the value of the parameter fact_c_energy of my scenario ?

Additional information: a custom scenario has been injected into the prompt to simulate the scenario interaction between the AI and the user.

6.3.3.1 Using OpenAI embedding model and his embedded data QA version

The parameter "Energy scale price" refers to the price at which energy is sold or purchased. It is a factor that affects the cost of energy and can impact the overall cost of the scenario. The value of the parameter "fact_c_energy" in your scenario is 5.

6.3.3.2 Using OpenAI embedding model and his embedded data QA version with source

The parameter "Energy scale price" refers to the price of energy in the scenario. It is not mentioned in the provided scenario. The value of the parameter "fact_c_energy" in the scenario is 1. /// SOURCES /// Energyscope

6.3.3.3 Using open-source embedding model and his embedded data QA version

The parameter "Energy scale price" refers to the price of energy in the scenario. It is not provided in the given scenario inputs.

6.3.3.4 Using open-source embedding model and his embedded data QA version with source

The parameter "Energy scale price" refers to the price of energy in the scenario. It is not mentioned in the provided scenario. The value of the parameter "fact_c_energy" in the scenario is 5. /// SOURCES ///

6.3.4 Question 3: I want to make my scenario more ecological, can you advice me on how to do it based on the independent and neutral scenario ?, I also wanted to know if you can give me information about C02 emission in Switzerland

6.3.4.1 Using OpenAI embedding model and his embedded data QA version

To make your scenario more ecological based on the independent and neutral scenario, you can consider the following actions:

1. Increase the share of renewable energy sources (such as solar and wind) in your electricity generation mix. This will reduce the reliance on fossil fuels and decrease CO2 emissions.

2. Implement energy efficiency measures in buildings, industries, and appliances to reduce energy consumption and lower CO2 emissions.

3. Promote the use of electric vehicles (EVs) and public transportation to reduce emissions from transportation.

4. Encourage the use of sustainable and low-carbon heating options, such as heat pumps, in buildings.

5. Increase the share of biomass and other renewable sources in the heating and cooling sector.

Regarding CO2 emissions in Switzerland, I don't have access to real-time data. However, Switzerland has been taking various measures to reduce CO2 emissions, including promoting renewable energy, energy efficiency, and sustainable transportation. You can refer to official government sources or environmental organizations for the latest information on CO2 emissions in Switzerland.

6.3.4.2 Using OpenAI embedding model and his embedded data QA version with source

To make your scenario more ecological, you can consider the following measures based on the independent and neutral scenario:

1. Increase the share of renewable energy sources such as solar, wind, hydro, and geothermal power.
2. Implement energy efficiency measures in buildings, industries, and appliances to reduce energy consumption.
3. Promote the use of electric vehicles and public transportation to reduce emissions from transportation.
4. Encourage the use of sustainable and low-carbon fuels in transportation, such as biofuels and

hydrogen.
5. Implement waste management strategies to reduce waste generation and promote recycling and composting.

Regarding CO2 emissions in Switzerland, I don't have the specific information. However, you can refer to the Swiss Federal Office for the Environment (FOEN) or the Swiss Energy Perspectives report for detailed data on CO2 emissions in Switzerland.

/// SOURCES /// Energyscope

6.3.4.3 Using open-source embedding model and his embedded data QA version

Based on the provided scenario parameters, here are some suggestions to make your scenario more ecological:

1. Increase the share of renewable energy sources: You can increase the values of "elec_re_pv", "elec_re_wind", "elec_re_hd", "elec_re_hr", and "elec_re_geo" to promote the use of solar, wind, hydro, and geothermal energy. This will reduce the reliance on fossil fuels and decrease CO2 emissions.

2. Decrease energy consumption: You can reduce the values of "e_industry", "e_appliances", and "e_lighting" to promote energy efficiency in industry, appliances, and lighting. This will lower energy demand and reduce environmental impact.

3. Promote sustainable transportation: Increase the values of "share_bev" to encourage the use of electric vehicles and reduce emissions from transportation. You can also consider increasing the values of "share_public_transport" and "share_freight_transport_train" to promote the use of public transportation and rail freight, which are more sustainable options.

4. Optimize building energy efficiency: Decrease the value of "q_building" to promote better insulation and energy-efficient building designs. This will reduce energy consumption for heating and cooling.

Regarding CO2 emissions in Switzerland, the provided information does not include specific data on CO2 emissions. However, by implementing the above suggestions, you can significantly reduce CO2 emissions in your scenario by promoting renewable energy and energy efficiency.

6.3.4.4 Using open-source embedding model and his embedded data QA version

I'm sorry, but I couldn't find any information about making a scenario more ecological or CO2 emissions in Switzerland based on the given scenario and sources. /// SOURCES ///

6.4 Evaluation of their score

6.4.1 Detailed overview

Additional information: to calculate the score of the response time I have taken the fastest response as the benchmark by setting it at 10 and then rated the other responses times proportionately.

MODEL \ CRITERION (MAX 10 PER COLUMNS)	RESPONSE QUALITY	RESPONSE TIME	ACCURACY
OPENAI - EMBEDDING AND DATASET PUSH	9	7 (38 sec)	9
OPEN-SOURCE - EMBEDDING AND DATASET PUSH	8	5 (83 sec)	8
OPENAI - QA VERSION (Q1)	8	10 (7.8 sec)	8
OPENAI - QA VERSION WITH SOURCE (Q1)	9	10 (7.8 sec)	10
OPEN-SOURCE - QA VERSION (Q1)	7	7.4 (10.5 sec)	7
OPEN-SOURCE - QA VERSION WITH SOURCE (Q1)	8	7.4 (10.5 sec)	9
OPENAI - QA VERSION (Q2)	8	9.2 (8.41 sec)	8
OPENAI - QA VERSION WITH SOURCE (Q2)	7	9.2 (8.41 sec)	9
OPEN-SOURCE - QA VERSION (Q2)	7	9.2 (8.40 sec)	8
OPEN-SOURCE - QA VERSION WITH SOURCE (Q2)	8	9.2 (8.40 sec)	9
OPENAI - QA VERSION (Q3)	9	4.2 (18.47 sec)	9
OPENAI - QA VERSION WITH SOURCE (Q3)	9	4.2 (18.47 sec)	10
OPEN-SOURCE - QA VERSION (Q3)	7	6.2 (12.40 sec)	7
OPEN-SOURCE - QA VERSION WITH SOURCE (Q3)	6	6.2 (12.40 sec	7

Figure 57 Evaluation of the answers from the 2 versions

Source: author

6.4.2 Evaluation summary

MODEL	TOTAL POINTS
OPENAI - EMBEDDING AND DATASET PUSH	25/30
OPEN-SOURCE - EMBEDDING AND DATASET PUSH	21/30
OPENAI - QA VERSION (SUM ACROSS Q1, Q2, Q3)	79.2/90
OPENAI - QA VERSION WITH SOURCE (SUM ACROSS Q1, Q2, Q3)	80.2/90
OPEN-SOURCE - QA VERSION (SUM ACROSS Q1, Q2, Q3)	64.8/90

OPEN-SOURCE - QA VERSION WITH SOURCE (SUM ACROSS Q1, Q2, Q3)	65.8/90
TOTAL FOR OPENAI MODELS	184.4/210
TOTAL FOR OPEN-SOURCE MODELS	151.6/210

Figure 58 evaluation summary

Source: author

MODEL	TOTAL RESPONSE QUALITY	TOTAL RESPONSE TIME	TOTAL ACCURACY
OPENAI	68	50.4	65
OPEN-SOURCE	59	45.2	56

Figure 59 total of points for each embedding model

Source: author

6.5 Matrix of comparison detailed

6.5.1 Response quality

6.5.1.1 OpenAI

Based on the score above, we can conclude that OpenAI embedding model is helping the AI model better understand the context of the user prompt, it provides more accurate context and information the AI model.

6.5.1.2 Open-source

Beside his speed, the open-source model isn't as helpful as the OpenAI embedding model, he was less able to help the LLM output qualitive and accurate text.

6.5.2 Accuracy

6.5.2.1 OpenAI

Same as the response quality section, OpenAI outperformed the Open source embedding model by 9 points. Accurate answers are crucial in this project since we are talking about Energyscope and energy related domains.

6.5.2.2 Open-source

Unfortunately, the Open-source model also lost the accuracy criterion.

6.5.3 Speed

6.5.3.1 OpenAI

While the OpenAI text embedding model offers greater accuracy and more contextual information than its open-source counterpart, it also operates at a slower pace. The complexity of Question 3 highlighted this difference particularly well, demonstrating that the response time increases significantly when the AI model tackles more complex inquiries.

6.5.3.2 Open-source

By being locally available, the open-source model naturally outperformed the OpenAI embedding model by 5.2 points.

6.5.4 Privacy & Security

6.5.4.1 OpenAI

Like the GPT-3,5 model, according to OpenAI publication on June 14 on their policies section, starting on March 01 2023, OpenAI will not use anymore the data submitted by users from the API to train and improve theirs models unless the users specifically tell them to do it. Nevertheless, any data sent trough the API will be stored for 30 days for abuse and misuse monitoring (*API Data Privacy*, n.d.).

6.5.4.2 Open-source

Local and open-source, the privacy and security is 100% under control of the user.

6.5.5 Cost

6.5.5.1 OpenAI

As discussed on the chapter 1.9.3.3 “pricing of text-embedding”, using the text embedding model comes with a cost of 3000 pages per 1 CHF. (*Pricing*, n.d.)

6.5.5.2 Open-source

Local and open source, the privacy and security is 100% under control of the user

6.5.6 Scalability

6.5.6.1 OpenAI

As it is hosted on OpenAI servers, the embedding model is highly scalable despite is slower score for the speed (*OpenAI Platform*, n.d.).

6.5.6.2 Open-source

The open-source model, being locally hosted, is reliant on the resources of my machine. This local dependence makes it less scalable compared to OpenAI (*FAISS & Sentence Transformers: Fast Semantic Search | Towards Data Science*, n.d.).

6.6 Overview Matrix

MODEL	RESPONSE QUALITY	ACCURACY	SPEED	PRIVACY AND SECURITY	COST	SCALABILITY
OPENAI	✓	✓	✗	✓	✗	✓
OPEN-SOURCE	✗	✗	✓	✓	✓	✗

Figure 60 matrix of comparison for the embedding models
Source: author

6.7 Choice: OpenAI

I decided to go with OpenAI embedding model “text-embedding-ada-002”, the bachelor project needs a high-quality embedding that can provide additional context and information to the AI model. Given the complexity of Energyscope and energy-related topics, it's crucial to employ the most superior embedding model available. (Yu, 2023)

Although the OpenAI embedding model comes with a slower processing speed and isn't free of cost, its high scalability makes it a worthwhile investment. It ensures a seamless accommodation of future data influxes that require embedding, avoiding potential integration issues down the line.

7 Project

The project goal is the following: implement artificial intelligence into the Energyscope tool and make it interactable about the user scenario, the user can also interact with the AI about Energyscope knowledge and energy related topics by pressing on the “wiki_mode” checkbox. I decided to go with Pinecone for the knowledge storage and LangChain for the connection between Pinecone and the LLM (OpenAI). An embedding model will be used for the project.

I will explain from the data gathering part to implementation of the AI in Energyscope and point out the problems and bugs encountered during the project. At the end I will also provide the potential solutions and feedback about the result of the performance & efficacity of the AI.

7.1 Why did we ignore finetuning

Initially, fine-tuning was my chosen approach for this project. However, after extensive testing and research, I concluded that fine-tuning wasn't the most suitable method for this particular project. Despite this, I decided to retain the entire section on fine-tuning to illustrate its potential applications and his specific use case. (Barreto, 2022)

Finetuning couldn't deliver what the combination of Langchain, Pinecone, and GPT-3.5 was capable of providing. A finetuned model is restricted to a specific task and requires a large number of generation pairs to comprehend the patterns of a new task.

7.2 Data gathering

The first part of the project was gathering the data to create a database of knowledge that will be used by the AI. The knowledge can be divided into 2 parts

- Knowledge about Energy scope and energy in general
- Knowledge about the different scenarios and their parameters

The knowledge about Energyscope and energy in general has been extracted from multiples sources such as:

- Master thesis
- Research papers
- Articles

Those documents will help the Artificial intelligence to better understand the questions related to

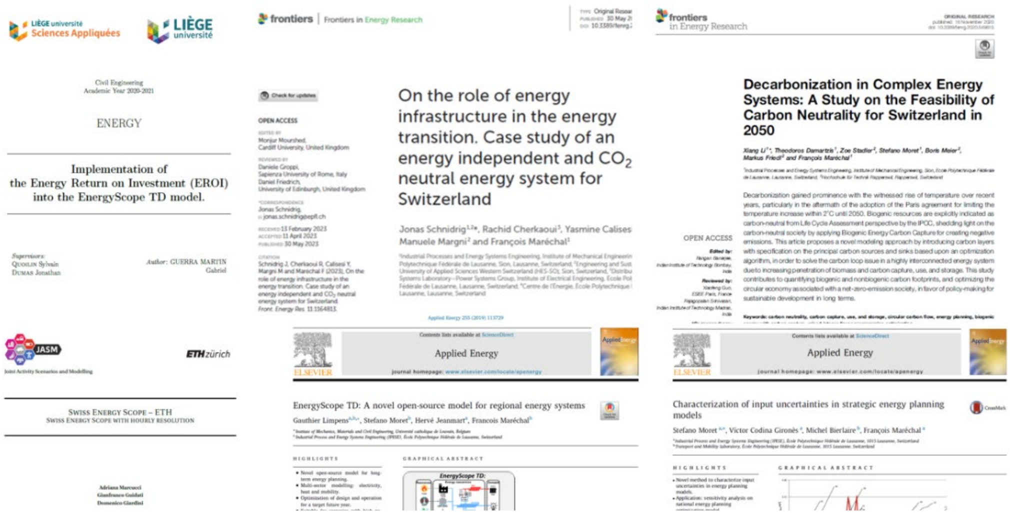


Figure 61 examples of documents used for the dataset

Sources: https://infoscience.epfl.ch/record/282912?ln=en /https://infoscience.epfl.ch/record/303482?ln=en /https://infoscience.epfl.ch/record/302978?ln=en energy and Energyscope in general (Li, 2022), (Moret, 2017), (Codina Gironès, 2018), (Schnidrig, 2018), (Schnidrig et al., 2023), (Fischer, 2023), (Schnidrig et al., 2023), (Souttre, 2022), (Slaymaker, 2021), (Brun, 2022), (Schnidrig, 2020), (Li et al., 2021), (Chuat, 2023), (Chuat, 2022), (Briguet, 2022), (Schnidrig et al., 2021), (Schnidrig et al., 2021), (Schnidrig et al., 2022), (Mathieu, 2022), (Schnidrig et al., 2021).

Those documents contains information's about different topics related to energy such as:

- Energy laws and policies
- IPCC reports on climate changes
- Methodologies
- Wallis Energyscope
- Perspectives for hydropower in Switzerland
- Mapping synergies and trade-offs between energy and the Sustainable Development

The data used for helping the AI understand the scenarios created by the user has been extracted from the Energyscope project available on Gitlab (*Files · Master · Ipese / EnergyScope / User Interfaces / Calculator.Energyscope.Ch · GitLab*, 2023). The following document used are the following :

- en_ini.pdf
  - Used for explaining parameters by giving their definitions
    - share_ind_boiler : Boiler
    - share_ind_elec : Electric heater
    - pop: Number of inhabitants in Switzerland.
- inputsRef.pdf
  - Used to specify the values of each parameters for the template's scenarios
    - Name of param | scenario template | activity|value
      - elec_nre_ccs | 2020 | 1 | 0
  - Scenarios such as:
    - 2050 Plus
    - Independent & Neutral
    - 2020
- inputs.pdf
  - Used for explaining the category and subcategory for each parameters
  - I also added the name of the parameter that is used on energyscope to help the AI better link the slider name to the ampl_param
    - Energy price to fact_c_energy
  - The document will also help the AI by specifying the unit used and the min max values for each parameters.
- res_description.pdf
  - Used for explaining the definition of each result values, it will help the AI better understand on which values to take when being questioned about improving the user

scenario.

- Indirect - Indirect employment
- res_gwp - Total CO2-equivalent emissions
- res_e_final - Final energy demand

7.3 Data preprocessing

7.3.1 Processing the data related to energy knowledge.

The first action that I took when pushing the pinecone to the cloud was to take the pdfs and transform them into text. The problem with this method is that conversion from a PDF to a raw text was badly made for CSV files, the characters where not converting properly and the result was often disappointing.

7.3.1.1 Transforming PDFs into texts

The libraries used for dealing with the pdf documents are the following

- Tiktoken
  - Used for the tokenisation of text (D, 2023)
- Pypdf
  - Used for pdf manipulation and transformation (*Welcome to Pypdf — Pypdf 3.13.0 Documentation*, n.d.)
- Openai
  - Used for implementing OpenAi API (*OpenAI Platform*, n.d.)
- Pinecone-client
  - Used for implementing pinecone connection and accessing / pushing the data (*Python Client*, n.d.)
- LangChain
  - Used for doing the bridge between OpenAI API and pinecone (Chase, 2022/2022)

```
from langchain.document_loaders import PyPDFLoader
```

I needed to import first the tools needed to perform the processing, PyPDFLoader for loading its content into memory and manipulating the PDF (*PDF | 🦜🔗 Langchain*, n.d.).

```
loader = PyPDFLoader("en_ini.pdf")

data = loader.load()

print (f'You have {len(data)} document(s) in your data')
```

The pdf is loaded into the loader variable and loaded into the data variable by calling the method .load() the data is now containing the content of the PDF file.
The print statement is used to check if the document has been loaded properly, if a PDF file contains

9 pages it will print out "9 documents". The content of the pdf is now ready to be transformed into embedded text.

For the embedding I used the OpenAI embedding model "text-embedding-ada-002", it is the most suited model for my project since it is low cost and performant (*OpenAIEmbeddings | 🦜🔗 Langchain*, n.d.).

embeddings = OpenAIEmbeddings(openai_api_key=OPENAI_API_KEY)

The embedding variable is instantiated and given the embedding model of OpenAI. This variable will be used by pinecone for embedding the text before sending it to the database.

pinecone.init(

api_key='4b440d15-634c-4a77-8ca1-fc8ade2ffaa6',

environment='eu-west4-gcp'

)

index_name = "energyscope7"

The instantiation of pinecone is done by calling the method .init() from the pine-cone client library (*Quickstart*, n.d.), the inputs needed is the pinecone API key and the environment name, the index_name variable contains the name of the index used for the project.
The name of the index is available on the index page by clicking on "connect" at the top right of the screen.

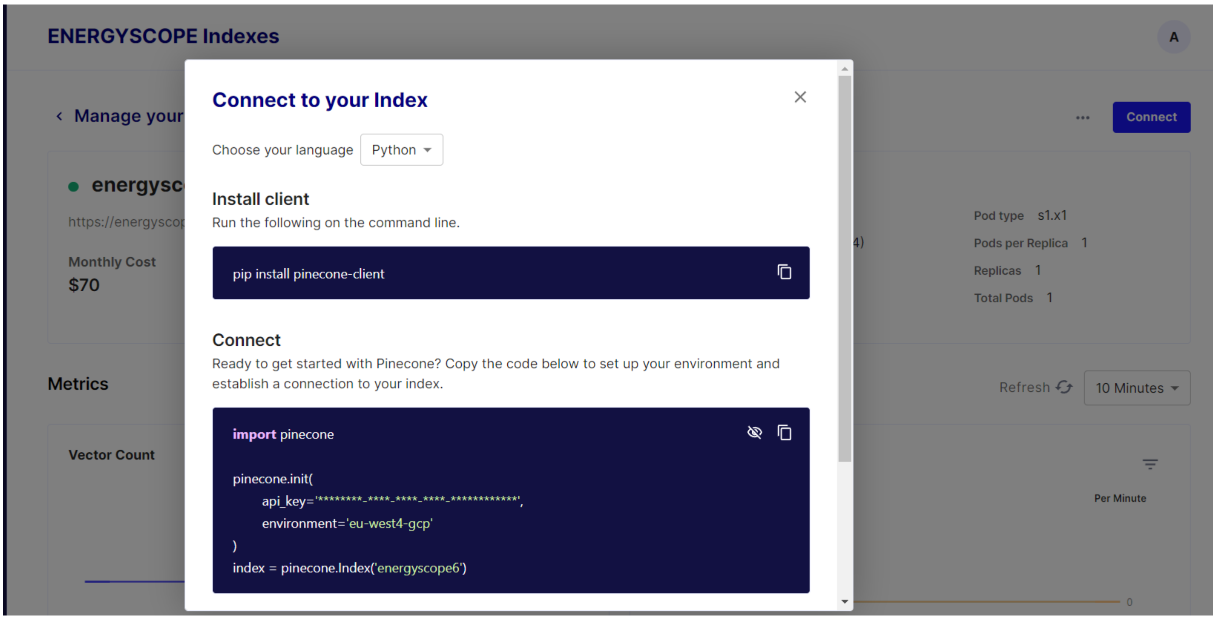


Figure 62 pinecone, index connection
Source: https://www.pinecone.io/

docsearch = Pinecone.from_texts([t.page_content for t in texts], embeddings, index_name=index_name)

By calling the method from_texts from the Pinecone class, we are embedding the text and pushing it into the pinecone database to the specified index.
You can check into the pinecone website if the data has been pushed successfully.

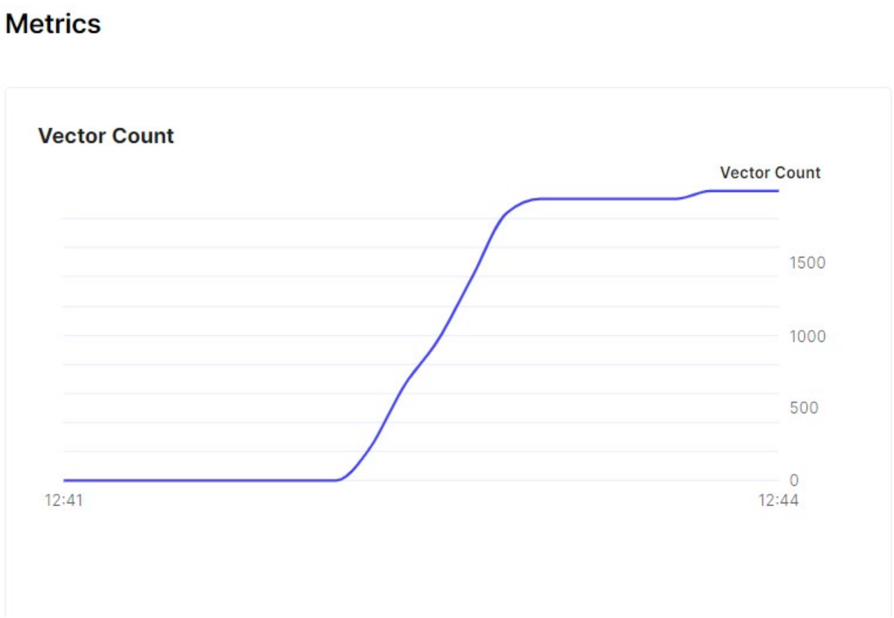


Figure 63 vector count after the data push

Source: https://www.pinecone.io/

7.3.1.2 Setback 1: PDF conversion with image recognition

As I was extracting the text from the PDF with the Pypdf library I saw that some part of the text was not converted as it should, the problem occurs when the document is complex, contains mathematical blocks, images, graphs, or special characters. Consequently, I decided to test the Tesseract library for some of these instances. Unfortunately, it didn't deliver satisfactory results either (*Tesseract OCR*, 2014/2023), (*Reading Text from the Image Using Tesseract - GeeksforGeeks*, n.d.).

PDF converted with Pypdf	PDF converted with Tesseract (Image recognition)
1 Introduction 1 1.1 Background . . . . 1 1.2 State of the art . . . . 3 1.3 Problem Statement 3	1 Introduction 1.1 Background .... 0... .. ee 1.2 Stateoftheart.. 2... .... ee 1.3 ProblemStatement ... 2... 0.0.0.0... cc ee 1.3.1 Research questions ...... 0.0... ee ee 1.3.2 Objectives 2... eee 1.4 Project Background... .. 2... 0... ee

1.3.1 Research questions . . . . 3 1.3.2 Objectives . . . . 3 1.4 Project Background . . . . 4 2 Modelization of the price of fossil fuels 5 2.1 Description of the model 5	 2 Modelization of the price of fossil fuels 2.1 Descriptionofthemodel .... 00. cee ee

Figure 64 difference between PDF converted with Pypdf and Tesseract

Source: author

Based on the poor result given with image recognition I sticked with the PyPDF library, to reduce the conversion anomalies (coming from CSV files) I directly opened the CSV and saved in a txt format to increase the quality of the output instead of using a python script to convert it. The solution is not very automated but since there wasn't a lot of CSV I didn't find a particular reason to find a solution to automate this.

7.3.2 Processing the data related to Energyscope.

The data used for helping the AI understand the Energyscope tool are the CSV files available in the data folder in the Energyscope Gitlab (*Files · Master · Ipese / EnergyScope / User Interfaces / Calculator.Energyscope.Ch · GitLab*, 2023).

The CSV were directly converted from the software itself to avoid any loss of data quality.

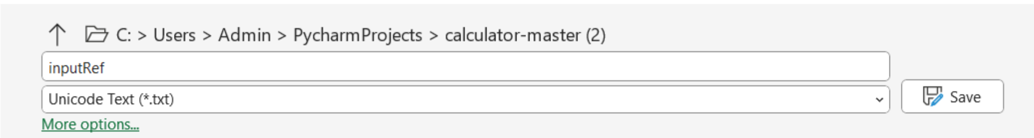


Figure 65 saving csv into text

Source: author

7.3.3 Creation of the dataset

To enable the sources citation functionality, you have to create a custom dataset should at least have 2 columns:

- Content column
  - Used to store the knowledge that will be used by the AI
- Source column
  - Used by the AI to know what the source of the documents is.

Other columns are optional for the good functioning of sources citations.
One of the main difficulty that i encountered during the creation of the dataset is that I couldn't fit

a full document into a single excel cell, due to its limitation of 32767 characters, when I tried to insert a document that was bigger than the limit allowed the software deleted the cell or sliced it across multiples cells therefore breaking the structure of the dataset (*Document QA | 🦜🔗 Langchain*, n.d.).

Final result of the dataset

id	url	title	text
1	https://www.sciencedirect.com/s	EnergyScope TD: A novel open-sou	Contentslistsavailableat ScienceDirect
2	https://www.sciencedirect.com/s	EnergyScope TD: A novel open-sou	fthespecificemissions(gwp
3	https://www.sciencedirect.com/s	EnergyScope TD: A novel open-sou	triesconnectedtoeachother.Models,suchas
4	https://www.sciencedirect.com/s	EnergyScope TD: A novel open-sou	teriaDecisionAnalysis(MCDA)forloadprofilingapplications.Appl
5	https://www.sciencedirect.com/s	Characterization of input uncertai	Characterization of input uncertainties in strategic energy
6	https://www.sciencedirect.com/s	Characterization of input uncertai	in parallel of the ï¬ve criteria to each uncertain parameter in a
7	https://www.sciencedirect.com/s	Characterization of input uncertai	he early 2000s. The uncertainty range
8	https://www.sciencedirect.com/s	Characterization of input uncertai	f Swiss nuclear power plants. 1990â€“2014 data
9	https://upcommons.upc.edu/hand	Implementation of the Energy Ret	Civil Engineering
10	https://upcommons.upc.edu/hand	Implementation of the Energy Ret	leted geological deposits could lead to a decrease in EROI. At

Figure 66 dataset ready to be pushed

Source: author

To avoid this problematic I created a script that transforms a PDF into text and then split it into multiples rows if the numbers of characters exceed the limit (*Maximum Characters Count in an Excel Cell | XlsIO | Syncfusion*, n.d.).

Data exceeding cell limit

id	url	title	text
1	Energysco	en_ini	[calculato
2	Energysco	dataEnergyScope	
3	Energysco	dataEnergyScope	large-
4	Energysco	res_description	{"inputs":
5	Energysco	en_ini	
6	Energysco	en_ini	category
A			
	Energysco	en_ini	res_e_fin
NA	Energysco	en_ini	category

Figure 67 : dataset structure when the character limit is exceeded.

Source: author

```
MAX_CHARACTERS = 30000
def add_entry_to_csv(title, source, file_path, csv_file_path="datasetComparison.csv"):
  # Check the file extension
  if file_path.endswith('.pdf'):
    # Open the PDF file
    with open(file_path, 'rb') as pdf_file:
      pdf = pypdf.PdfReader(pdf_file)
      text = ''
      # Read the text from each page
      for page in pdf.pages:
        text += page.extract_text()
  elif file_path.endswith('.txt'):
    # Open the text file and read the content
    with open(file_path, 'r', encoding='utf-8', errors='ignore') as txt_file:
      text = txt_file.read()
  # Split text into chunks
  text_chunks = [text[i:i + MAX_CHARACTERS] for i in range(0, len(text), MAX_CHARACTERS)]
  # Open the CSV file
  with codecs.open(csv_file_path, 'a', 'utf-8') as csv_file:
    # Create a CSV writer
    csv_writer = csv.writer(csv_file)
    # If the CSV file is empty, write the header
    if os.stat(csv_file_path).st_size == 0:
      csv_writer.writerow(["id", "url", "title", "text"])
    # Get the last id in csv
    try:
      last_id = list(csv.reader(codecs.open(csv_file_path, 'r', 'utf-8')))[-1][0]
      if last_id == '':
        last_id = 0
    except IndexError:  # CSV is empty
      last_id = 0
```

The function add_entry_to_csv takes as input:

- Title
  - The title of the document.
- Source
  - The url source of the document.
- File_path
  - The file path where the document is stored.
- CSV_file_path
  - The csv file path where the function will store the text.

The function check first the extension of the given document and apply the correct conversion to the document.

The text is first divided into segments of 30,000 characters to prevent surpassing the cell restriction. Subsequently, the CSV is accessed, and if it does not exist, it will be established. The headers are then placed into the CSV. Following this, data rows are inputted into the CSV, with the id also being incremented during the process.

Example of PDF processing

add_entry_to_csv (title='A modelling framework for assessing the impact of green mobility technologies on energy systems',source='https://infoscience.epfl.ch/record/288731?ln=en', file_path='pdfs/paper_112.pdf')

add_entry_to_csv(title='Appendix - A modelling framework for assessing the impact of green mobility technologies on energy systems',source='https://infoscience.epfl.ch/record/287243?ln=en',file_path='pdfs/ECOS_2021___Modelling_Framework_Mobility (1).pdf')

7.3.4 Pushing the dataset into Pinecone

The dataset is now ready, we can push it to the vector database, we will then use this database to help the AI understand better the context of our prompt by giving him additional knowledge. To do this we will need the following code.

The libraries used for this part are:

- Tiktoken
  - Used to encode text into tokens (D, 2023)
- LangChain
  - RecursiveCharacterTextSplitter
    - Used for splitting the “text” column of

each rows (*RecursiveCharacterTextSplitter | 🦜🔗 Langchain*, n.d.)

- openai.OpenAIEmbeddings
  - Used to get the embedding model from OpenAI (*OpenAIEmbeddings | 🦜🔗 Langchain*, n.d.)

- Pandas
  - Used for data manipulation, specially CSV in my case (*Pandas - Python Data Analysis Library*, n.d.)
- Tdqm
  - Optional, used to display a progress bar loop for progress tracking (*Tqdm · PyPI*, n.d.)
- Uuid
  - Used to generate unique identifiers for each text chunk (oaclaf, n.d.)
- Pinecone
  - Used for instantiating the connection to the pinecone vector database
  - It is where we will store the “knowledge” of the AI (*Pinecone*, n.d.)
- Datasets
  - Used for providing easy access to a wide variety of dataset on HuggingFace.com, contains tools for data loading, transformation and pre-processings (*Datasets*, n.d.)

```
#DATASET
df = pd.read_csv('datasetV13.csv')
df = df.astype(str)
# Convert pandas dataframe to HuggingFace dataset
data = Dataset.from_pandas(df)
#PINECONE
index_name = 'energyscope12800'
PINECONE_API_KEY = ‘CONFIDENTIAL’
PINECONE_ENVIRONMENT = 'eu-west4-gcp'
#OPENAI
model_name = 'text-embedding-ada-002'
tiktoken.encoding_for_model('gpt-3.5-turbo-16k')
tokenizer = tiktoken.get_encoding('cl100k_base')
```

We will need 3 things, the dataset that we created earlier, the pinecone instance with his API key and his environment, and OpenAI embedding model. The dataset is converted to a huggingface dataset avoid compatibility issue. The model used for the embedding is the text-embedding-ada-002, it is at the moment the best embedding model available at OpenAI.

MODEL	ROUGH PAGES PER DOLLAR	PERFORMANCE
TEXT-EMBEDDING-ADA-002	3000	53.9
-DAVINCI--001	6	52.8
-CURIE--001	60	50.9
-BABBAGE--001	240	50.4
-ADA--001	300	49.0

Figure 68 embedding models available at OpenAI

Source: https://openai.com/pricing

```
def tiktoken_len(text):
    tokens = tokenizer.encode(
        text,
        disallowed_special=()
    )
    return len(tokens)
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=700,
    chunk_overlap=20,
    length_function=tiktoken_len,
    separators=["\n\n", "\n", " ", ""]
)
# get openai api key from platform.openai.com
embed = OpenAIEmbeddings(
    model=model_name,
    openai_api_key=OPENAI_API_KEY
)
pinecone.init(
    api_key=PINECONE_API_KEY,
    environment=PINECONE_ENVIRONMENT
)
```

We first create the function tiktoken_len() that take as input a text and return the length of tokens, it will be used by the RecursiveCharacterTextSplitter to help him determine the length of a text string in terms of the number of tokens it contains when encoded by the specified tokenizer (*RecursiveCharacterTextSplitter | 🦜️🔗 Langchain*, n.d.).

- Chunk_size
  - Determine the maximum size of each chunk that is created by the text splitter
- Chunk_overlap
  - Specifies how much overlap there should be between 2 consecutive chunk of text
  - Used to be sure that no context is lost when splitting the text.
- Length_function
  - Used for determining the length in tokens of each pieces of text.
- Separators
  - If not set, the splitter will cut the text by the specified chunk size, the separators help the splitter by telling the "natural" boundaries or the text and therefore avoiding the splitting in the middle of a word.

The embed variable is the instance of the OpenAI embedding model and is set by specifying the model's name and the OpenAI API key. The pinecone.init method instantiate the connection to the pinecone vector database by giving it the pinecone API key and the environment of the pinecone project (Available on the pinecone website).

```
if index_name  not in pinecone.list_indexes():
  # we create a new index
  pinecone.create_index(
    name=index_name,
    metric='cosine',
    dimension=1536  # 1536 dim of text-embedding-ada-002
  )
index = pinecone.GRPCIndex(index_name)
```

Before pushing the data, we must first be sure that the index that we work with exist, if the index does not exist in our pinecone project it will create a new one and therefore setting the index name (*Create_index Pinecone*, n.d.).

```
for i, record in enumerate(tqdm(data)):
    # first get metadata fields for this record
    metadata = {
        'wiki-id': str(record['id']),
        'source': record['url'],
        'title': record['title']
    }
    # now we create chunks from the record text
    record_texts = text_splitter.split_text(record['text'])
    # create individual metadata dicts for each chunk
    record_metadatas = [{
        "chunk": j, "text": text, **metadata
    } for j, text in enumerate(record_texts)]
    # append these to current batches
    texts.extend(record_texts)
    metadatas.extend(record_metadatas)
    # if we have reached the batch_limit we can add texts
    if len(texts) >= batch_limit:
        ids = [str(uuid4()) for _ in range(len(texts))]
        embeds = embed.embed_documents(texts)
        index.upsert(vectors=zip(ids, embeds, metadatas))
        texts = []
        metadatas = []
if len(texts) > 0:
    ids = [str(uuid4()) for _ in range(len(texts))]
    embeds = embed.embed_documents(texts)
    index.upsert(vectors=zip(ids, embeds, metadatas))
```

The final part is the processing, chunking of the dataset. The loop goes trough each record in the dataset. The enumerate function is used to get the index of each record. When looping into each record we create a dictionary of metadata containing the id, url source, title of the record. The metadata will be used by the AI to know what the source of a specific record is. Each chunk of text of the record will be assigned a dictionary metadata that contains information about his record.

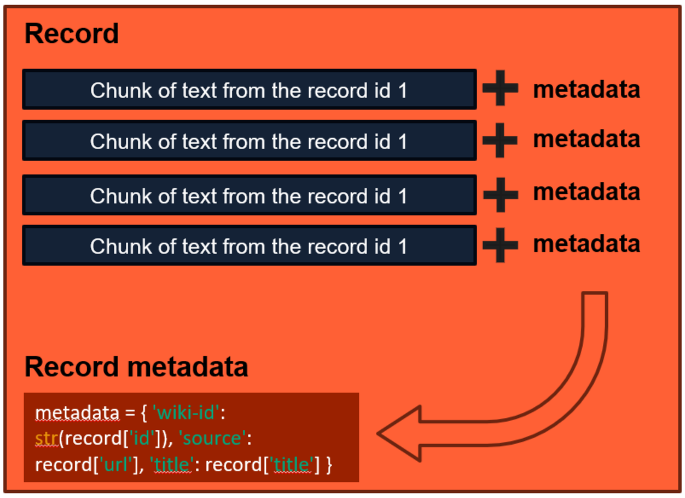


Figure 69 representation of the metadata

Source: author

When the batch limit is exceed, the chunks of texts are then embedded using the embed_documents() method and then pushed to the index by using index.upsert().

7.4 Implementation of Q&A with sources about general knowledge

The implementation of the Q&A has been done by connecting the vector database with OpenAI GPT-3.5 using LangChain framework. The process start first by initializing the embedding model using the method OpenAiEmbeddings() available from the LangChain library (*Document QA | 🦜🔗 Langchain*, n.d.).

```
from langchain.embeddings  import OpenAIEmbeddings
embed = OpenAIEmbeddings(
        model=model_name,
        openai_api_key=OPENAI_API_KEY
       )
```

The method takes as input the embedding model name (text-embedding-ada-002) and the OpenAI API key. The API key can be used for the embedding model and the GPT-3.5, you do not need separate keys for each.

Then the initialization of the connection the Pinecone vector database is done (*Pinecone*, n.d.).

```
import pinecone
pinecone.init(
        api_key=PINECONE_API_KEY,
        environment=PINECONE_ENVIRONMENT
    )
```

It is done by calling the method init() and specifying the API key and the environment name of the vector database.

After that the vector store must be initialised to allow semantic search based on the prompt provided.

```
from langchain.vectorstores import Pinecone
import pinecone
index = pinecone.Index(index_name)
vectorstore = Pinecone(
                index, embed.embed_query, text_field
                )
vectorstore.similarity_search(
                finalPrompt,
                k=10
                )
```

The pinecone.index() method is used because the vector store does not accept string as index name. Therefore, this method is employed to fetch the specific index ID corresponding to the provided name.

The vector store is initialized by using the class Pinecone from the LangChain library, it will create a Pinecone object by providing the name of the index, the function used to embed a query, the text_field is used to specify the column name that contains the text content of the documents.

The large language model must also be instantiated using the LangChain class ChatOpenAi.

```
from langchain.chat_models import ChatOpenAI
llm = ChatOpenAI(
        openai_api_key=OPENAI_API_KEY,
        model_name='gpt-3.5-turbo-16k',
        temperature=0.0
    )
```

The llm variable is ChatOpenAI object that will take as input the OpenAI key, the model name used, the reason why the GPT-3.5 16k model is used is because the total number or tokens exceed 12'000 if we inject also the user scenario and therefore we need a model capable of handling more than this amount of tokens.

And now the most important part, the linking of everything we instantiated before. To do this we need to create a RetrievalQAWithSourcesChain.from_chain_type() chain from the LangChain library (*Langchain.Chains.Qa_with_sources.Retrieval.RetrievalQAWithSourcesChain — 🦜🔗 LangChain 0.0.241*, n.d.).

```
from langchain.chains import RetrievalQAWithSourcesChain
qa_with_sources = RetrievalQAWithSourcesChain.from_chain_type(
        llm=llm,
        chain_type="stuff",
        retriever=vectorstore.as_retriever()
    )
responseRaw = qa_with_sources(finalPrompt)
```

The chain is now created, we can start to play with the Q&A model.

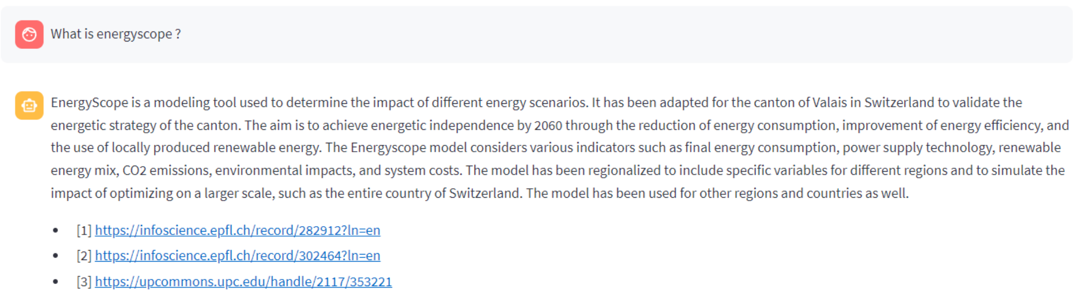


Figure 70 chat with Q&A with sources

Source: author

7.4.1 Before & After knowledge injection

7.4.1.1 Question 1: What is Energyscope ?

Without knowledge injection

I don't know what energyscope is.
/// SOURCES ///

With knowledge injection

EnergyScope is an open-source energy model used for urban and regional energy system planning. It is designed to assess future energy scenarios and can evaluate cases with high penetration rates of renewable energy sources, storage needs, and innovative technologies. EnergyScope has been used in various studies, including the Belgian Energy Transition and the Swiss Energy Transition. It allows for the coordination of distributed investments and centralized actors in promoting a feasible energy transition. The model is freely available and has a monthly to hourly time resolution. It incorporates mobility and heat supply in addition to the electricity sector. EnergyScope has been adapted for specific regions, such as the canton of Valais in Switzerland. It can be used to validate and optimize regional energy strategies and assess the impact of different scenarios on indicators such as energy consumption, renewable energy mix, CO2 emissions, environmental impacts, and system costs.
/// SOURCES ///
- [34] Jeroen DOMMISSE, Jean-Louis TYCHON, Modelling of Low Carbon Energy Systems for 26 European Countries with EnergyScopeTD
- [35] Gauthier Limpens, Hervé Jeanmart 1 and Francois Maréchal, Belgian Energy Transition: What Are the Options?
- [36] Règlement de Gouvernance, PROJET DE PLAN WALLON ENERGIE CLIMAT 2030
- [37] Danielle Devogelaer, Jan Duerinck, Dominique Gusbin, et al., Towards 100% renewable energy in Belgium by 2050
- https://upcommons.upc.edu/handle/2117/353221
- https://infoscience.epfl.ch/record/302464?ln=en
- https://www.sciencedirect.com/science/article/pii/S0306261919314163
- https://infoscience.epfl.ch/record/282912?ln=en

7.4.1.2 Question 2: What will be the energy plan of Valais Switzerland in 2050 ?

Without knowledge injection

I don't know the answer to this question.
/// SOURCES ///

With knowledge injection

The energy plan of Valais, Switzerland in 2050 is outlined in the report "Valais 2050" by the Canton of Valais. The report discusses the energy conversion in Valais and compares it to the Swiss Energy Strategy 2.0 (SES 2.0) for Switzerland as a whole. The report analyzes different scenarios and their implications for carbon neutrality and the optimal energy system. The report concludes that the energetic strategy of Valais is feasible and aligns with the suggested solution given by the Canton of Valais. The optimal solution for Valais results in a minimization of total cost for the region. However, it is noted that the optimal solution for Valais differs from the optimal solution for Switzerland, particularly in terms of electricity export. The report also discusses the use of different energy sources, such as hydro power, solar thermal, waste, wet biomass, wind, and wood. The cost comparison shows that the total cost for the Valais 2050 scenario is more than twice as high as the VSES 2050 scenario for Switzerland. The report emphasizes the importance of considering the role of Valais as part of Switzerland in the energy planning process.
/// SOURCES ///
- https://infoscience.epfl.ch/record/282912?ln=en
- https://www.frontiersin.org/articles/10.3389/fenrg.2020.549615/full

7.4.1.3 Question 3: What is the parameter energy price scale and why is it useful?

Without knowledge injection

I don't know the answer to this question.
/// SOURCES ///

With knowledge injection

The parameter energy price scale refers to the cost of energy production and consumption. It is a crucial factor in determining the overall cost of energy. The energy price scale is useful because it helps in evaluating the economic feasibility of energy projects and assessing the competitiveness of different energy sources. It also plays a significant role in energy planning and decision-making processes.
///SOURCES///
Energyscope,
https://www.sciencedirect.com/science/article/pii/S0306261917306116?via%3Dihub

7.4.1.4 Evaluation of the score

QUESTION	WITHOUT KNOWLEDGE INJECTION (0-10)	WITH KNOWLEDGE INJECTION (0-10)

Q1: UNDERSTANDING OF ENERGYSCOPE	0	10
Q2: KNOWLEDGE OF VALAIS, SWITZERLAND'S 2050 ENERGY PLAN	0	10
Q3: UNDERSTANDING OF THE PARAMETER ENERGY PRICE SCALE	0	10

Figure 71 comparison before and after knowledge injection

Source: author

Without additional knowledge, the AI model isn't capable of understanding what we are talking about, the model had no clue what Energyscope was, wasn't able to answers questions about the energy plan of Valais in 2050 and of course couldn't understand questions related to the scenario configuration.

7.5 Implementation of Q&A on the user scenario and comparison with other scenario templates.

In order to contrast the user-defined scenario with the various predefined template scenarios, it was necessary to incorporate into the vector database, information relating to the scenarios, including:

- An overview and description of each scenario
- Specific scenario templates, such as:
  - Independent & neutral
  - 2050
- Detailed explanations of each parameter within these scenarios, providing:
  - A definition of the parameter
  - An explanation of its relevance and utility in the scenario

For more information about which documents I used please refer to the chapter 8.1 "Data gathering". To add the scenario to the prompt I retrieved the scenario that is given my the Energyscope tool and inserted it the following way:

query = "QUESTION : What is the parameter energy price scale and why is it useful ?"

query += "INSTRUCTION only use the following scenario if you cannot answer without it, here is my scenarios parameters values : "+json.dumps(scenario).__str__()

The final prompt is a result of merging both pieces of data: the initial prompt and the newly integrated scenario data retrieved from the JSON format. This ensures all relevant information is included in a consolidated and comprehensive prompt.

7.5.1 Analysis of the feature

Additional information: The following analysis was done on the class object provided by LangChain, one is able to cite sources while the other cannot.

Q&A version

```
qa = RetrievalQA.from_chain_type(
  llm=llm,
  chain_type="stuff",
  retriever=vectorstore.as_retriever()
)
response = qa (prompt)
```

Q&A with source version

```
qa_with_sources  = RetrievalQAWithSourcesChain.from_chain_type(
   llm=llm,
   chain_type="stuff",
   retriever=vectorstore.as_retriever()
)
response = qa_with_sources(prompt)
```

Figure 72 chat showing the comparison of scenarios

Source:author

7.5.1.1 Question 1 What do you suggest making my scenario more green?

Q&A version

To make your scenario more green, you can consider the following suggestions:

1. Increase the share of renewable energy sources such as solar and wind power. This can be done by increasing the values of "elec_re_pv" and "elec_re_wind" parameters in your scenario.

2. Reduce the reliance on fossil fuel imports by decreasing the value of "elec_import" parameter.

3. Implement energy efficiency measures in buildings, industries, and appliances. This can be achieved by increasing the values of "q_building", "e_industry", "e_appliances", and "e_lighting" parameters.

4. Promote the use of electric vehicles and public transportation. You can increase the values of "share_bev" and "share_public_transport" parameters to encourage the adoption of electric vehicles and public transportation.

5. Explore the potential of other renewable energy sources such as geothermal energy. You can increase the value of "elec_re_geo" parameter to utilize geothermal energy.

6. Encourage the use of sustainable waste management practices. You can increase the value of "share_q_waste" parameter to promote waste recycling and reduce waste generation.

Q&A with sources version

To make the scenario more green, you can consider the following parameters:
1. Increase the value of "elec_re_pv" (solar energy) and "elec_re_wind" (wind energy) to generate more renewable energy.
2. Increase the value of "share_bev" (share of electric vehicles) to promote the use of electric vehicles and reduce emissions from transportation.
3. Increase the value of "share_q_wood" (share of wood as a fuel) to promote the use of renewable biomass energy.
4. Decrease the value of "fact_c_energy" (carbon factor for energy) to reduce carbon emissions from energy production.
5. Increase the value of "fact_c_invest" (carbon factor for investment) to prioritize investments in green technologies and infrastructure.
6. Increase the value of "fact_i_rate" (interest rate) to incentivize investments in green projects.
7. Decrease the value of "elec_import" (imported electricity) to reduce reliance on non-renewable

energy sources.
8. Decrease the value of "co2_limit" (CO2 emissions limit) to set stricter emissions targets.

Please note that these suggestions are based on the given parameters and may vary depending on specific circumstances and goals.
/// SOURCES /// Energyscope

7.5.1.2 Question 2: What should I change if I want my scenario to match the independent & neutral scenario?

Q&A version

To match the independent & neutral scenario, you should change the following parameters in your scenario:

1. Set the value of "fact_c_energy" to 1. //correct
2. Set the value of "fact_c_invest" to 1. // good value already
3. Set the value of "fact_i_rate" to 2.1. // good value already
4. Set the value of "elec_import" to 0. // correct
5. Set the value of "ng_import" to 0.0. // good value already
6. Set the value of "co2_limit" to 0. // good value already
7. Set the value of "elec_re_pv" to 41.0. // correct
8. Set the value of "elec_re_wind" to 5.5. // correct
9. Set the value of "elec_re_hd" to 8.2. // good value already
10. Set the value of "elec_re_hr" to 4.5. // correct
11. Set the value of "elec_re_geo" to 0.7. // correct

Q&A with sources version

To match the independent and neutral scenario, you would need to change the following parameters in your scenario:

- Set "share_public_transport" to 0 (active: true) // not correct
- Set "share_freight_transport_train" to 0 (active: true) // not correct
- Set "share_ind_cogen" to 0 (active: true) // not correct
- Set "share_ind_boiler" to 0 (active: true) // not correct
- Set "share_ind_elec" to 0 (active: true) // not correct
- Set "share_dhn" to 0 (active: true) // not correct
- Set "share_dec" to 0 (active: true) // not correct
- Set "share_dhn_hp" to 0 (active: true) // not correct

- Set "share_dhn_cogen" to 0 (active: true) // not correct
- Set "share_dhn_boiler" to 0 (active: true) // not correct
- Set "share_dhn_deep_geo" to 0 (active: true) // not correct
- Set "share_dec_ehp" to 0 (active: true) // not correct
- Set "share_dec_thp" to 0 (active: true) // not correct
- Set "share_dec_cogen" to 0 (active: true) // not correct
- Set "share_dec_acogen" to 0 (active: true) // not correct
- Set "share_dec_boiler" to 0 (active: true) // not correct
- Set "share_dec_solar" to 0 (active: true) // not correct
- Set "share_dec_elec" to 0 (active: true) // not correct
- Set "share_q_ng" to 0 (active: true) // not correct
- Set "share_q_wood" to 0 (active: true) // not correct
- Set "share_q_lfo" to 0 (active: true) // not correct
- Set "share_q_waste" to 0 (active: true) // not correct
- Set "share_q_coal" to 0 (active: true) // not correct

/// SOURCES /// Energyscope

7.5.1.3 Question 3: What is the pop value of my scenario?

Q&A version

The pop value of your scenario is 10.

Q&A with sources version

The population value of the given scenario is not provided in the extracted parts of the document.
/// SOURCES ///

7.5.1.4 Evaluation of the score

QUESTION	Q&A VERSION SCORE (0-10)	Q&A WITH SOURCES VERSION SCORE (0-10)
Q1: SUGGESTIONS TO MAKE THE SCENARIO GREENER	10	10
Q2: CHANGES TO MATCH THE INDEPENDENT & NEUTRAL SCENARIO	9	0
Q3: THE POPULATION VALUE OF THE USER'S SCENARIO	10	0

Figure 73 evaluation of the Q&A and the Q&A with sources
Source: author

Q&A version evaluation

The AI model responded with accuracy for the question 1 by providing detailed advice on how to become more green. In the question 2, the response given was also good, although a few of the suggestions were not particularly useful, the model was successful in delivering 6 appropriate tips from the list. Lastly, in response to question 3, the AI model accurately identified the value of a user scenario parameter, something the Q&A with sources version failed to do.

Q&A with sources version evaluation

The AI model performed very well on the question 1, he was able to respond with accuracy which parameters to modify in order to make the user scenario greener. In the question 2, the model answered by suggesting modifying certain parameters of the user scenario, the problem is that the values suggested are all at zero, which is not at all the right answer. In response to the third question, the model was unable to provide the value of the population parameter for the user's scenario, despite having this information available in the prompt.

7.5.2 Q&A with sources limitation

From the evaluation matrix, it's evident that the Q&A with sources version struggled with comparing user scenarios. This shortcoming stems from its primary design purpose, which is to answer questions directly based on provided documents and also to reference them. This design is why it performs well with straightforward Q&A tasks but underperforms in comparative questions.

For questions that involve comparisons, the standard Q&A version is more fitting. It demonstrates better accuracy and understanding of the values in a user's scenario.

One problem that is facing the sources version is that when injecting into the prompt the scenario, the AI model is unable to respond to simple Q&A questions, seems like the scenario is blurring the understanding of the question.

To overcome this problem a checkbox named “Wiki_Mode” has been added so the user can specifiy if he wants to do a comparison or general Q&A with the AI. If the checkbox is checked it will switch to the Q&A with sources version and only the question will be sent if unchecked we will use the Q&A version and send also the scenario config (*Streamlit Docs*, n.d.).

```
if wiki_mode:
    # If WikiMode is enabled, use the RetrievalQAWithSourcesChain  model
    qa = RetrievalQAWithSourcesChain.from_chain_type(
        llm=llm,
        chain_type="stuff",
        retriever=vectorstore.as_retriever()
    )
else:
    # If WikiMode is disabled, use the RetrievalQA model
    qa = RetrievalQA.from_chain_type(
        llm=llm,
        chain_type="stuff",
        retriever=vectorstore.as_retriever()
    )
```

7.6 Implementation of Chat session

The chat session is a feature that saves the precedent interaction with the AI model to avoid repeating the entire context for each question. The context window length will be at 16k tokens, with the scenario injection the prompt is already at 12k meaning the user will have 4k tokens available for his chat history (*Memory* | 🦜 🔗 *Langchain*, n.d.).

LangChain provide chat memory modules called

- ChatMessageHistory
- ConversationBufferMemory

I could have taken those modules and implement it for handling the sessions but i went with a easier solution, working with streamlit sessions (Eden Marco, 2023).

To add this functionality, I added 3 session state:

```
if "messages" not in st.session_state:
    st.session_state.messages = [{'role': 'assistant', 'content': 'Ask me anything about Energyscope !'}]

if "wiki_mode_session" not in st.session_state:
    st.session_state.wiki_mode_session = wiki_mode

if 'scenario' not in st.session_state:
    st.session_state.scenario = None
```

The session state "messages" will store the messages conversations of the user and the AI. The wiki_mode_session will store the value of the checkbox wikimode and finally the scenario will store the whole scenario of the user (*Session State - Streamlit Docs*, n.d.).

7.6.1 When to update the chat history

During the initialisation of the website, the scenario is stored in the session state and the value of the wikimode checkbox also. We store them because we need to know when we must delete the entire chat history.

In our case the chat history must be deleted when:

- User make a change in the scenario
- Click on the "wiki_mode" checkbox

To do this, at each refresh we will check if one of those 2 points have changed.

```
current_scenario = json.dumps(configuration_state)  # or whatever variable represents the current scenario
if st.session_state.wiki_mode_session != wiki_mode or st.session_state.scenario != current_scenario:
    st.session_state.messages = [{'role': 'assistant', 'content': 'Ask me anything about Energyscope !'}]
    st.session_state.wiki_mode_session = wiki_mode
    st.session_state.scenario = current_scenario
```

By doing this way, we will ensure that if any changes have been to the scenario or if the wiki mode checkbox has changed his value, the chat would be deleted.

7.6.2 Adding the conversations into the chat history

After receiving the answers from the AI, the prompt and the response are stored into the “messages” variable.

```
st.session_state.messages.append({"role": "user", "content": prompt})
st.session_state.messages.append({"role": "assistant", "content": responseClean})
```

We store the messages in this structure, so it is easier to differentiate who was the emitter of the message and it is much easier for looping the messages.

```
for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])
```

By doing this loop, it displays the chat history visually into the Energyscope website.

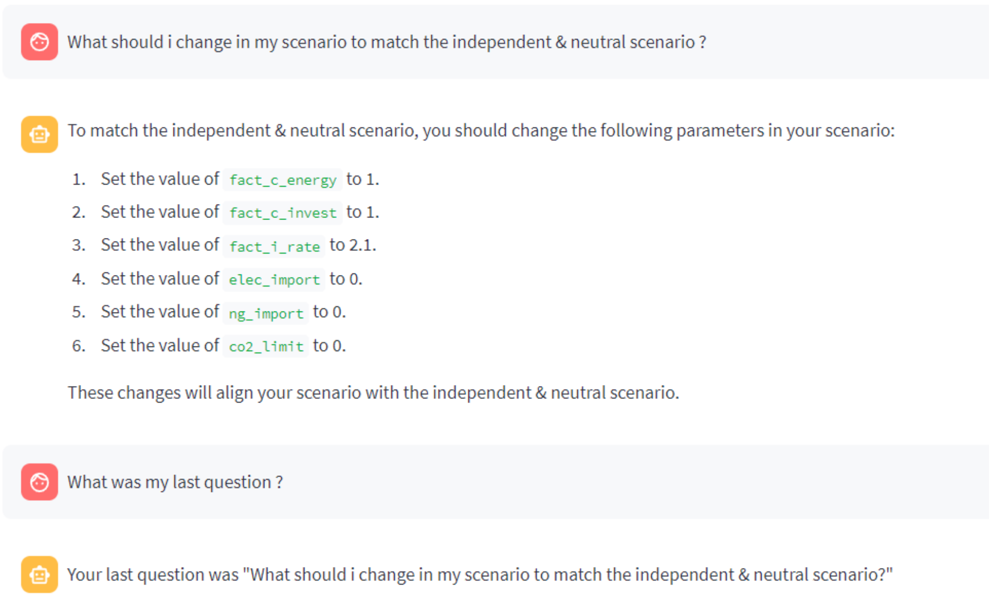


Figure 74 chat showing the memory feature
Source: author

7.7 Current limitations encountered

7.7.1.1 Limitation 1: Lack of understanding how the parameters work with each others

While testing the AI model, I frequently noticed that the AI struggled to comprehend the interconnections among the variables. Although it recognized their importance, it failed to fully grasp how they functioned collectively.

7.7.1.2 Limitation 2: Lack of understanding on the generation of scenario values

When questioned why the given value was 45, the AI was unable to provide a satisfactory answer. This indicates that the model lacks insight into the process behind the computation of these values.

7.8 Potential solutions

7.8.1 Integrate explanatory documents regarding scenario generation.

To mitigate the issue of the AI's inadequate comprehension on the generation of scenario values, one possible solution could be to incorporate into the vector database additional documents that explain the calculation methodology for each "res_" variable.

7.8.2 Incorporate a glossary clarifying each parameter and its interconnections.

The problem stemming from the AI's lack of understanding of the parameter relationships could be efficiently corrected by integrating documents that provide detailed explanations of each parameter, its purpose and how it impacts other parameters and outcomes.

8 Conclusion

Incorporating artificial intelligence into the Energyscope tools enhances the user's comprehension of the tool and provides insights into energy-related topics concerning Switzerland based on the research papers that we pushed on the vector databases.

Despite some limitations, I believe the integration still contributes positively to the overall value of the Energyscope tool. It enriches the user experience and assists him in creating more effective scenarios.

Despite the challenges faced, I found the project utterly fascinating to undertake. Given my conviction that AI will play an enormous role in the contemporary world, I opted for this topic.

9 References

🔑 *Getting Started | Chroma*. (n.d.). Retrieved July 19, 2023, from https://www.trychroma.com/getting-started

🤗 *Hub client library*. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/huggingface_hub/index

🤗 *PEFT*. (2023). [Python]. Hugging Face. https://github.com/huggingface/peft (Original work published 2022)

🤗 *Transformers*. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/index$

4 Reasons for Artificial Intelligence (AI) Project Failure in 2023. (n.d.). Retrieved May 30, 2023, from https://research.aimultiple.com/ai-fail/

7. Unsupervised Learning: Dimensionality Reduction - Machine Learning and Data Science Blueprints for Finance [Book]. (n.d.). Retrieved July 22, 2023, from https://www.oreilly.com/library/view/machine-learning-and/9781492073048/ch07.html

A conversation with an artificial intelligence (AI). (2022, December 6). https://exmachina.ch/tech/a-conversation-with-an-artificial-intelligence/

A Gentle Introduction to 8-bit Matrix Multiplication for transformers at scale using transformers, accelerate and bitsandbytes. (n.d.). Retrieved July 22, 2023, from https://huggingface.co/blog/hf-bitsandbytes-integration

A guide to optimizing Transformer-based models for faster inference | Tryolabs. (n.d.). Retrieved July 23, 2023, from https://tryolabs.com/blog/2022/11/24/transformer-based-model-for-faster-inference

A new mechanism for freezing extra dimensions with higher-order curvature terms—ScienceDirect. (n.d.). Retrieved June 7, 2023, from https://www.sciencedirect.com/science/article/pii/S0370269320306602

About | LMSYS Org. (n.d.). Retrieved May 24, 2023, from https://lmsys.org/about

Accelerate. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/accelerate/index

Ahmad. (2021, April 8). *Cannot import name “T5Tokenizer” from “transformers.models.t5”* [Forum post]. Stack Overflow. https://stackoverflow.com/q/66998668

AI Limitations in 2023: Data hungry, opaque, brittle systems. (n.d.). Retrieved July 22, 2023, from https://research.aimultiple.com/ai-limitations/

AI Model: How Does It Work? (2022, August 30). Koombea. https://www.koombea.com/blog/ai-model/

Alpaca: The Large Language Model That Won't Fleece You. (n.d.). Hackster.Io. Retrieved July 23, 2023, from https://www.hackster.io/news/alpaca-the-large-language-model-that-won-t-fleece-you-cd133fec7412

Amirjavid, F. (2021, July 28). *Answer to "cannot import name 'T5Tokenizer' from 'transformers.models.t5.'"* Stack Overflow. https://stackoverflow.com/a/68559561

An important next step on our AI journey. (2023, February 6). Google. https://blog.google/technology/ai/bard-google-ai-search-updates/

An Introduction to Using Transformers and Hugging Face. (n.d.). Retrieved July 23, 2023, from https://www.datacamp.com/tutorial/an-introduction-to-using-transformers-and-hugging-face

Andermann, T., Antonelli, A., Barrett, R. L., & Silvestro, D. (2022). Estimating Alpha, Beta, and Gamma Diversity Through Deep Learning. *Frontiers in Plant Science*, *13*. https://www.frontiersin.org/articles/10.3389/fpls.2022.839407

Anunaya, S. (2021, August 10). Data Preprocessing in Data Mining—A Hands On Guide (Updated 2023). *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/08/data-preprocessing-in-data-mining-a-hands-on-guide/

API data privacy. (n.d.). Retrieved July 24, 2023, from https://openai.com/api-data-privacy

Artificial Intelligence Data Limitations for Organizations | InfoClutch. (2023, April 17). https://www.infoclutch.com/installed-base/artificial-intelligence-data-limitations/

Asgarinejad, F. (2019, May 22). *Answer to "What are tokens and tokenizations?"* Data Science Stack Exchange. https://datascience.stackexchange.com/a/52369

Ask AI: Why does OpenAI use 1,536 dimensions for embeddings (specifically the text-embedding-ada-002 model)? (n.d.). Retrieved July 23, 2023, from https://www.theinternet.io/articles/ask-ai/why-does-openai-use-1536-dimensions-for-embeddings-specifically-the-text-embedding-ada-002-model/

Augmented Startups (Director). (2017). *Support Vector Machine (SVM) in 7 minutes—Fun Machine Learning*. https://www.youtube.com/watch?v=Y6RRHw9uN9o

Auto Classes. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/model_doc/auto

baeldung. (2023, March 23). *What Does Learning Rate Warm-up Mean? | Baeldung on Computer Science*. https://www.baeldung.com/cs/learning-rate-warm-up

Barreto, S. (2022, August 9). *What Is Fine-Tuning in Neural Networks? | Baeldung on Computer Science*. https://www.baeldung.com/cs/fine-tuning-nn

Batch mapping. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/datasets/about_map_batch

Best practices for prompt engineering with OpenAI API | OpenAI Help Center. (n.d.). Retrieved July 22, 2023, from https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api

Bhargav, N. (2022, August 25). *Neurons in Neural Networks | Baeldung on Computer Science*. https://www.baeldung.com/cs/neural-networks-neurons

Bhatt, S. (2019, April 19). *Reinforcement Learning 101*. Medium. https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292

Bigscience/bloom-7b1 · Hugging Face. (n.d.). Retrieved July 24, 2023, from https://huggingface.co/bigscience/bloom-7b1

Brdiczka, O. (n.d.). *Contextual AI: The Next Frontier of Artificial Intelligence*. Retrieved July 22, 2023, from https://business.adobe.com/blog/perspectives/contextual-ai-the-next-frontier-of-artificial-intelligence

Briguet, R. (Ed.). (2022). *Assessment of price decomposition and distribution of fossil fuels*.

Brownlee, J. (2018, December 2). A Gentle Introduction to Dropout for Regularizing Deep Neural Networks. *MachineLearningMastery.Com*. https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/

Brun, J. (Ed.). (2022). *Reducing Greenhouse Gas emissions is not the only solution*.

BTC Tweets Sentiment. (n.d.). Retrieved July 23, 2023, from https://www.kaggle.com/datasets/aisolutions353/btc-tweets-sentiment

Busquet, M. (2023, July 21). Fine-Tuning LLaMA 2 Models using a single GPU, QLoRA and AI Notebooks. *OVHcloud Blog*. https://blog.ovhcloud.com/fine-tuning-llama-2-models-using-a-single-gpu-qlora-and-ai-notebooks/

Caelen, O. (2023, February 18). *Unleashing the Power of GPT: How to Fine-Tune Your Model*. Medium. https://towardsdatascience.com/unleashing-the-power-of-gpt-how-to-fine-tune-your-model-da35c90766c4

Causal language modeling. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/tasks/language_modeling

Chains | 🦜🔗 LangChain. (n.d.). Retrieved July 24, 2023, from https://docs.langchain.com/docs/components/chains/

Chang, P. (2023, May 21). LLaMA-LoRA Tuner: UI tool to fine-tune and test your own LoRA models. *Medium*. https://medium.com/@zetavg/llama-lora-tuner-ui-tool-to-fine-tune-and-test-your-own-lora-models-704e2e645d5e

Chase, H. (2022). *LangChain* [Python]. https://github.com/hwchase17/langchain (Original work published 2022)

Chat with Open Large Language Models. (n.d.). Retrieved July 24, 2023, from https://gradio.app/

Chris Alexiuk (Director). (2023a). *Low-rank Adaption of Large Language Models: Explaining the Key Concepts Behind LoRA*. https://www.youtube.com/watch?v=dA-NhCtrrVE

Chris Alexiuk (Director). (2023b). *Low-rank Adaption of Large Language Models Part 2: Simple Fine-tuning with LoRA*. https://www.youtube.com/watch?v=iYr1xZn26R8

Chroma. (n.d.). Retrieved July 24, 2023, from https://www.trychroma.com/

Chroma | 🦜🔗 Langchain. (n.d.). Retrieved July 19, 2023, from https://python.langchain.com/docs/modules/data_connection/vectorstores/integrations/chroma

Chroma-langchain/persistent-qa.ipynb at master · hwchase17/chroma-langchain. (n.d.). Retrieved July 19, 2023, from https://github.com/hwchase17/chroma-langchain/blob/master/persistent-qa.ipynb

Chuat, A. (Ed.). (2022). *Application of levelized infrastructure-connected regionalisation in energy systems modelling*.

Chuat, A. (Ed.). (2023). *Impact of renewable energy hubs configurations on the national infrastructure*.

CJ Gammon (Director). (2022, November 30). *GPT 3 Model Fine-Tune Walkthrough*. https://www.youtube.com/watch?v=_RTN8CWFUsc

Cloud GPUs. (n.d.). Retrieved June 7, 2023, from https://cloud-gpus.com/

Code Understanding | 🦜🔗 Langchain. (n.d.). Retrieved July 19, 2023, from https://python.langchain.com/docs/use_cases/code/

Codina Gironès, V. (2018). *Scenario modelling and optimisation of renewable energy integration for the energy transition* [EPFL]. https://doi.org/10.5075/epfl-thesis-8780

Community | Pinecone. (n.d.). Retrieved July 24, 2023, from https://www.pinecone.io/community/

Components | 🦜🔗 LangChain. (n.d.). Retrieved July 24, 2023, from https://docs.langchain.com/docs/category/components

Could you train a ChatGPT-beating model for $85,000 and run it in a browser? (n.d.). Retrieved July 24, 2023, from https://simonwillison.net/2023/Mar/17/beat-chatgpt-in-a-browser/

Cox, J. (2023, May 25). *How the A.I. explosion could save the market and maybe the economy*. CNBC. https://www.cnbc.com/2023/05/25/how-the-ai-explosion-could-save-the-market-and-maybe-the-economy.html

Create_index Pinecone. (n.d.). Pinecone. Retrieved July 24, 2023, from https://docs.pinecone.io/reference/create_index

CUDA Toolkit 12.2 Downloads | NVIDIA Developer. (n.d.). Retrieved July 19, 2023, from https://developer.nvidia.com/cuda-downloads?target_os=Windows&target_arch=x86_64&target_version=11&target_type=exe_local

CUHK partners with Shanghai Artificial Intelligence Laboratory, SenseTime and Shanghai Jiao Tong University to unveil a new paradigm of general vision model INTERN | CUHK Communications and Public Relations Office. (n.d.). CUHK Partners with Shanghai Artificial Intelligence Laboratory, SenseTime and Shanghai Jiao Tong University to Unveil a New Paradigm of General Vision Model INTERN | CUHK Communications and Public Relations Office. Retrieved May 30, 2023, from https://www.cpr.cuhk.edu.hk/en/press/cuhk-partners-with-shanghai-artificial-intelligence-laboratory-sensetime-and-shanghai-jiao-tong-university-to-unveil-a-new-paradigm-of-general-vision-model-intern/

D, H. (2023, June 25). Tokenization In OpenAI API: Let's Explore Tiktoken Library. *Medium*. https://medium.com/@basics.machinelearning/tokenization-in-openai-api-lets-explore-tiktoken-library-d02d3ce94b0a

Data Collator. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/main/main_classes/data_collator

Datasets. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/datasets/index

David Shapiro ~ AI (Director). (2022). *5 Tips and Misconceptions about Finetuning GPT-3*. https://www.youtube.com/watch?v=VfAsu_dxw0g

Decapoda-research/llama-7b-hf · Hugging Face. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/decapoda-research/llama-7b-hf

DeepMind AI Reduces Google Data Centre Cooling Bill by 40%. (n.d.). Retrieved May 23, 2023, from https://www.deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-by-40

Demo, G.-3. (n.d.). *Gopher by DeepMind | Discover AI use cases*. Retrieved July 23, 2023, from https://gpt3demo.com/apps/deepmind-gopher

Department of Computer Science. (n.d.). Toronto Metropolitan University. Retrieved May 30, 2023, from https://www.torontomu.ca/cs/

Deployment | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://python.langchain.com/docs/guides/deployments/

Dezhic, E. (2017, July 28). *Generalization in AI Systems*. Medium. https://towardsdatascience.com/generalization-in-ai-systems-79c5b6347f2c

Discord—ChromaDB. (2023, July 22). Discord. https://discord.com/channels/1073293645303795742/1074571738525999164

Ditto. (2023, February 4). *GPT-3 vs. GPT-3.5: What's new in OpenAI's latest update?* Accubits Blog.

https://blog.accubits.com/gpt-3-vs-gpt-3-5-whats-new-in-openais-latest-update/

Document QA | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://python.langchain.com/docs/modules/chains/additional/question_answering

DocumentStore. (n.d.). Retrieved July 24, 2023, from https://docs.haystack.deepset.ai/docs/document_store

Doshi, K. (2021, June 3). *Transformers Explained Visually (Part 3): Multi-head Attention, deep dive*. Medium. https://towardsdatascience.com/transformers-explained-visually-part-3-multi-head-attention-deep-dive-1c1ff1024853

Doshi, S. (2020, August 3). *Various Optimization Algorithms For Training Neural Network*. Medium. https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6

D'Sa, A. G. (2021, October 28). *In the context of Deep Learning, what is training warmup steps* [Forum post]. Data Science Stack Exchange. https://datascience.stackexchange.com/q/55991

DSwithBappy (Director). (2022). *OpenAI GPT 3 in One Video | Fine Tuning GPT 3 | How to Use OpenAI API ?* https://www.youtube.com/watch?v=tA_ICy3H8H4

Eden Marco (Director). (2023). *Memory in LangChain | Deep dive (python)*. https://www.youtube.com/watch?v=70lqvTFh_Yg

Embeddings | Machine Learning. (n.d.). Google for Developers. Retrieved July 23, 2023, from https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture

EnPaceRequiescat. (2023, June 10). *What hardware do I need for fine tuning/training?* [Reddit Post]. R/LocalLLaMA. www.reddit.com/r/LocalLLaMA/comments/14639ol/what_hardware_do_i_need_for_fine_tuningtraining/

Essayez Bard, une expérience d'IA conversationnelle par Google. (n.d.). Retrieved July 23, 2023, from https://bard.google.com

evchaki. (2023, June 23). *Vector Database*. https://learn.microsoft.com/en-us/semantic-kernel/memories/vector-db

FAISS & Sentence Transformers: Fast Semantic Search | Towards Data Science. (n.d.). Retrieved July 24, 2023, from https://towardsdatascience.com/master-semantic-search-at-scale-index-millions-of-documents-with-lightning-fast-inference-times-fa395e4efd88

Faraday.dev. (n.d.). Retrieved May 30, 2023, from https://faraday.dev

FastChat. (2023). [Python]. LMSYS. https://github.com/lm-sys/FastChat (Original work published 2023)

Files · master · Ipese / EnergyScope / User interfaces / calculator.energyscope.ch · GitLab. (2023, May 26). GitLab. https://gitlab.epfl.ch/ipese/energyscope/user-interfaces/calculator/-/tree/master

Fine Tuning: OpenAI Models + Your Confluence Data. (2023, March 16). TrueFoundry Blog. https://blog.truefoundry.com/training-fine-tuning-of-llms-with-your-own-data/

Fine-tune LLaMA to speak like Homer Simpson. (2023, March 17). https://replicate.com/blog/fine-tune-llama-to-speak-like-homer-simpson

Fine-tuning a Neural Network explained. (n.d.). Retrieved July 22, 2023, from https://deeplizard.com/learn/video/5T-iXNNiwIs

Fine-tuning Alpaca and LLaMA: Training on a Custom Dataset | MLExpert - Crush Your Machine Learning interview. (n.d.). Retrieved May 30, 2023, from https://mlexpert.io/machine-learning/tutorials/alpaca-fine-tuning

Fine-tuning LLMs Made Easy with LoRA and Generative AI-Stable Diffusion LoRA | by xiao sean | Medium. (n.d.). Retrieved July 23, 2023, from https://xiaosean5408.medium.com/fine-tuning-llms-made-easy-with-lora-and-generative-ai-stable-diffusion-lora-39ff27480fda

Fine-Tuning OpenAI Models with Python: A Step-by-Step Guide. (n.d.). Articulate Python: Python Tutorials. Retrieved June 1, 2023, from https://www.articulatepython.com/blog/finetune-openai-models

Foundation Model AI. (2023, June 27). *Techopedia*. https://www.techopedia.com/definition/34826/foundation-model

freeCodeCamp.org (Director). (2020a). *Keras with TensorFlow Course—Python Deep Learning and Neural Networks for Beginners Tutorial*. https://www.youtube.com/watch?v=qFJeN9V1ZsI

freeCodeCamp.org (Director). (2020b, March 3). *TensorFlow 2.0 Complete Course—Python Neural Networks for Beginners Tutorial*. https://www.youtube.com/watch?v=tPYj3fFJGjk

freeCodeCamp.org (Director). (2022, October 6). *PyTorch for Deep Learning & Machine Learning –*

Full Course. https://www.youtube.com/watch?v=V_xro1bcAuA

Galav, A. (2022, April 11). *The Growing Demand for Artificial Intelligence (AI)*. Great Learning Blog: Free Resources What Matters to Shape Your Career! https://www.mygreatlearning.com/blog/the-growing-demand-for-artificial-intelligence-ai/

Ganesh, K. S. (2022, September 7). *What's The Role Of Weights And Bias In a Neural Network?* Medium. https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f

GanymedeNil/text2vec-cmedqq-lert-base · Hugging Face. (n.d.). Retrieved July 19, 2023, from https://huggingface.co/GanymedeNil/text2vec-cmedqq-lert-base

Gerganov, G. (2023). *Llama.cpp* [C]. https://github.com/ggerganov/llama.cpp (Original work published 2023)

Getting started with Haystack. (n.d.). Haystack Documentation. Retrieved July 19, 2023, from https://docs.haystack.deepset.ai/docs

Glossary. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/glossary

Google AI PaLM 2. (n.d.). Google AI. Retrieved July 23, 2023, from https://ai.google/discover/palm2/

Google, Apple and Meta have big AI plans to catch up with ChatGPT | Fortune. (n.d.). Retrieved July 23, 2023, from https://fortune.com/2023/02/03/google-meta-apple-ai-promises-chatgpt-earnings/

Google Colaboratory. (n.d.). Retrieved July 23, 2023, from https://colab.research.google.com/drive/1rqWABmz2ZfolJOdoy6TRc6YI7d128cQO?usp=sharing#scrollTo=VxB6UV5XAvvP

Google DeepMind. (n.d.). Retrieved May 14, 2023, from https://www.deepmind.com/

Google I/O 2023 unveils PaLM 2 large language model. (2023, May 10). Mashable. https://mashable.com/article/google-io-2023-palm2-ai-announcement

Google Trains 280 Billion Parameter AI Language Model Gopher. (n.d.). InfoQ. Retrieved July 23, 2023, from https://www.infoq.com/news/2022/01/deepmind-gopher/

GOYAL, C. (2021, July 19). 2023's Best Guide to Discriminative & Generative Machine Learning Models.

Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/07/deep-understanding-of-discriminative-and-generative-models-in-machine-learning/

GPT-1 to GPT-4: Each of OpenAI's GPT Models Explained and Compared. (2023, April 11). MUO. https://www.makeuseof.com/gpt-models-explained-and-compared/

Gpt2 · Hugging Face. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/gpt2

GPT-2: 1.5B release. (n.d.). Retrieved July 23, 2023, from https://openai.com/research/gpt-2-1-5b-release

GPT-3 powers the next generation of apps. (n.d.). Retrieved July 23, 2023, from https://openai.com/blog/gpt-3-apps

GPT-3.5. (n.d.). Lab Lab. Retrieved July 23, 2023, from https://lablab.ai/tech/openai/gpt3-5

GPT-3.5 vs GPT-4—Here's what we know so far? (2023, May 11). VideoGamer.Com. https://www.videogamer.com/tech/ai/chatgpt-3-5-vs-chatgpt-4/

GPT-4. (n.d.). Retrieved July 23, 2023, from https://openai.com/research/gpt-4

GPT-4: How to use the AI chatbot that puts ChatGPT to shame. (2023, July 19). Digital Trends. https://www.digitaltrends.com/computing/chatgpt-4-everything-we-know-so-far/

Greyling, C. (2022, November 30). I Tested The New OpenAI GPT-3 Davinci Model. *Medium*. https://cobusgreyling.medium.com/i-tested-the-new-openai-gpt-3-davinci-model-df290f660d9d

Greyling, C. (2023a, May 3). How To Fine-Tune GPT-3 For Custom Intent Classification. *Medium*. https://cobusgreyling.medium.com/how-to-fine-tune-gpt-3-for-custom-intent-classification-95973d05d7e0

Greyling, C. (2023b, June 13). Creating A Custom Fine-Tuned Model With OpenAI's GPT-3 Language API. *Medium*. https://cobusgreyling.medium.com/creating-a-custom-fine-tuned-model-with-openais-gpt-3-language-api-a847364548b6

Gupta, K. (2023, February 25). Meta AI Unveils LLaMA: A Series of Open-Source Language Models Ranging from 7B to 65B Parameters. *MarkTechPost*. https://www.marktechpost.com/2023/02/25/meta-ai-unveils-llama-a-series-of-open-source-language-models-ranging-from-7b-to-65b-parameters/

Hardware Recommendations for Machine Learning / AI. (n.d.). Puget Systems. Retrieved July 24, 2023, from https://www.pugetsystems.com/solutions/scientific-computing-workstations/machine-learning-ai/hardware-recommendations/

Hardware spec for finetuning >7B Llama · Issue #160 · OptimalScale/LMFlow. (n.d.). GitHub. Retrieved July 24, 2023, from https://github.com/OptimalScale/LMFlow/issues/160

Haystack. (n.d.). Pinecone. Retrieved July 19, 2023, from https://docs.pinecone.io/docs/haystack

Haystack · A resilient, scalable tracing and analysis system. (n.d.). Retrieved July 24, 2023, from https://expediadotcom.github.io/haystack/

Haystack Benchmarks. (n.d.). Haystack Retrieved July 24, 2023, from https://haystack.deepset.ai/benchmarks/

Haystack Introduction. (n.d.). Haystack Documentation. Retrieved July 24, 2023, from https://docs.haystack.deepset.ai/docs

Hilbig, N. (2021, August 9). How Far Can Artificial Intelligence Go? The 8 Limits of Machine Learning. *CodeX*. https://medium.com/codex/how-far-can-artificial-intelligence-go-the-8-limits-of-machine-learning-383dd9b2f7bd

How is LLaMa.cpp possible? (n.d.). Retrieved July 23, 2023, from https://finbarr.ca/how-is-llama-cpp-possible/

How to evaluate a completion(QA) model? - Prompting. (2023, March 22). OpenAI Developer Forum. https://community.openai.com/t/how-to-evaluate-a-completion-qa-model/114074

How to use Alpaca-LoRA to fine-tune a model like ChatGPT. (2023, March 23). https://replicate.com/blog/fine-tune-alpaca-with-lora

https://www.facebook.com/48576411181. (n.d.). *7 Revealing Ways AIs Fail—IEEE Spectrum*. Retrieved May 30, 2023, from https://spectrum.ieee.org/ai-failures

https://www.googlecloudcommunity.com/gc/user/viewprofilepage/user-id/402219. (2023, March 29). *Google Bard API*. https://www.googlecloudcommunity.com/gc/AI-ML/Google-Bard-API/m-p/538517#M1526

Hugging Face – The AI community building the future. (2023a, May 29). https://huggingface.co/datasets

Hugging Face – The AI community building the future. (2023b, June 6). https://huggingface.co/

Hugging Face Hub | 🦜🔗 Langchain. (n.d.). Retrieved July 23, 2023, from https://python.langchain.com/docs/modules/model_io/models/llms/integrations/huggingface_hub

Huggingface/accelerate. (2023). [Python]. Hugging Face. https://github.com/huggingface/accelerate (Original work published 2020)

IBM Technology (Director). (2022, September 15). *Gradient Descent Explained*. https://www.youtube.com/watch?v=i62czvwDlsw

Importance of Neural Network Bias and How to Add It. (n.d.). Retrieved July 23, 2023, from https://www.turing.com/kb/necessity-of-bias-in-neural-networks

Inc, G. (n.d.). *tensorflow: TensorFlow is an open source machine learning framework for everyone*. (2.13.0) [Python]. Retrieved July 23, 2023, from https://www.tensorflow.org/

Inc, P. S. (n.d.). *pinecone-client: Pinecone client and SDK* (2.2.2) [Python; OS Independent]. Retrieved July 19, 2023, from https://www.pinecone.io

Inside the Pinecone | Pinecone. (n.d.). Retrieved July 24, 2023, from https://www.pinecone.io/blog/inside-the-pinecone/

Integrations | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://python.langchain.com/docs/integrations

Introducing ChatGPT. (n.d.). Retrieved July 23, 2023, from https://openai.com/blog/chatgpt

Introducing PaLM 2. (2023, May 10). Google. https://blog.google/technology/ai/google-palm-2-ai-large-language-model/

Introducing text and code embeddings. (n.d.). Retrieved July 23, 2023, from https://openai.com/blog/introducing-text-and-code-embeddings

Introducing Whisper. (n.d.). Retrieved July 23, 2023, from https://openai.com/research/whisper

Introduction | 🦜🔗 Langchain. (n.d.). Retrieved July 19, 2023, from https://python.langchain.com/docs/get_started/introduction

Introduction to Meta AI's LLaMA: Empowering AI Innovation. (n.d.). Retrieved July 23, 2023, from https://www.datacamp.com/blog/introduction-to-meta-ai-llama

Intuitive Machine Learning (Director). (2020). *Support Vector Machines: All you need to know!*

https://www.youtube.com/watch?v=ny1iZ5A8ilA

is 1536 dimension for embedding overkill? - Google Search. (n.d.). Retrieved July 19, 2023, from https://www.google.com/search?q=is+1536+dimension+for+embedding+overkill+%3F&rlz=1C1VDKB_deCH1045CH1045&ei=nUu1ZNeNF4yA9u8Pw-aN0AE&ved=0ahUKEwiX0KjI9ZWAAxUMgP0HHUNzAxoQ4dUDCBE&uact=5&oq=is+1536+dimension+for+embedding+overkill+%3F&gs_lp=Egxnd3Mtd2l6LXNlcnAiKmlzIDE1MzYgZGltZW5zaW9uIGZvciBlbWJlZGRpbmcgb3ZlcmtpbGwgP0jxJlDZFljWJXACeAGQAQCYAZgBoAGaD6oBBDAuMTS4AQPIAQD4AQHCAgoQABhHGNYEGLADwgIIECEYoAEYwwTiAwQYACBBiAYBkAYI&sclient=gws-wiz-serp

Is it permissible to use ChatGPT for Commercial Use? - ChatGPT. (2023, March 29). OpenAI Developer Forum. https://community.openai.com/t/is-it-permissible-to-use-chatgpt-for-commercial-use/128005

Johnson, D. (2023, May 27). *Unsupervised Machine Learning: Algorithms, Types with Example*. https://www.guru99.com/unsupervised-machine-learning.html

Kamalraj M M (Director). (2023). *Langchain VectorStores Show Down: Which One Reigns Supreme?* https://www.youtube.com/watch?v=zGAkhN1YZXM

Karanam, S. (2021, August 11). *Curse of Dimensionality—A "Curse" to Machine Learning*. Medium. https://towardsdatascience.com/curse-of-dimensionality-a-curse-to-machine-learning-c122ee33bfeb

Keras: Deep Learning for humans. (n.d.). Retrieved July 23, 2023, from https://keras.io/

Khatik, K. (2023, February 18). Mastering GPT-3: A Comprehensive Guide to Fine-Tuning with OpenAI, Complete with Examples. *Medium*. https://medium.com/@kapildevkhatik2/mastering-gpt-3-a-comprehensive-guide-to-fine-tuning-with-openai-complete-with-examples-e28515c22d92

Klingler, N. (2023, January 1). *The Ultimate Guide to Understanding and Using AI Models (2023)*. Viso.Ai. https://viso.ai/deep-learning/ml-ai-models/

Kozlov, D. (n.d.). *Bit Depth—Color precision in raster images*. Retrieved May 23, 2023, from https://www.the-working-man.org/2014/12/bit-depth-color-precision-in-raster.html

LaMDA: Our breakthrough conversation technology. (2021, May 18). Google. https://blog.google/technology/ai/lamda/

langchain.chains.qa_with_sources.retrieval.RetrievalQAWithSourcesChain—🦜🔗 LangChain

0.0.241. (n.d.). Retrieved July 24, 2023, from https://api.python.langchain.com/en/latest/chains/langchain.chains.qa_with_sources.retrieval.RetrievalQAWithSourcesChain.html

LangChainChromaStarter—Replit. (n.d.). Retrieved July 19, 2023, from https://replit.com/@swyx/LangChainChromaStarter#main.py

Language modelling at scale: Gopher, ethical considerations, and retrieval. (n.d.). Retrieved July 23, 2023, from https://www.deepmind.com/blog/language-modelling-at-scale-gopher-ethical-considerations-and-retrieval

Language Models. (n.d.). Haystack Documentation. Retrieved July 24, 2023, from https://docs.haystack.deepset.ai/docs

Latest Community topics. (n.d.). OpenAI Developer Forum. Retrieved July 24, 2023, from https://community.openai.com/c/community/21

Le Deep Learning de A à Z. (n.d.). Udemy. Retrieved May 31, 2023, from https://groupemutuel.udemy.com/course/le-deep-learning-de-a-a-z/

Li, X. (2022). *Towards a negative-emission society* [EPFL]. https://doi.org/10.5075/epfl-thesis-10041

Li, X., Muller, D., Schnidrig, J., & Maréchal, F. (Eds.). (2021). Application of artificial intelligence on uncertainty analysis for long-term energy system planning. *Proceedings of ECOS 2021*.

Liam Ottley (Director). (2023, January 26). *How to Fine Tune GPT3 | Beginner's Guide to Building Businesses w/ GPT-3*. https://www.youtube.com/watch?v=3EdEw4gyr-s

Lin, D. C.-E. (2023, April 10). *8 Simple Techniques to Prevent Overfitting*. Medium. https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d

LLaMA. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/main/model_doc/llama

LLaMA vs Alpaca: Comparing the Animal-Inspired AI Models. (n.d.). ProjectPro. Retrieved July 23, 2023, from https://www.projectpro.io/article/llama-vs-alpaca-models/866

llama.cpp vs llama - compare differences and reviews? | LibHunt. (n.d.). Retrieved July 23, 2023, from https://www.libhunt.com/compare-llama.cpp-vs-gmorenz--llama

Load. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/datasets/loading

Loading a Dataset. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/datasets/v1.11.0/loading_datasets.html

LocalLlama. (n.d.). Retrieved July 24, 2023, from https://www.reddit.com/r/LocalLLaMA/

Low-Rank Adaptation of Large Language Models (LoRA). (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/diffusers/training/lora

Lucidate (Director). (2023). *Fine-Tuning OpenAI*. https://www.youtube.com/watch?v=uFiI5fK-7B4

M, A. (2019, September 11). *Answer to "In the context of Deep Learning, what is training warmup steps."* Data Science Stack Exchange. https://datascience.stackexchange.com/a/60028

Mahmood, O. (2022, April 13). 🤗 *What's Hugging Face?* Medium. https://towardsdatascience.com/whats-hugging-face-122f4e7eb11a

Major Limitations of AI | Limitations of AI | Three Major Limitations of AI. (2020, February 19). https://apacentrepreneur.com/the-three-major-limitations-of-ai/

Malik, F. (2021, March 4). Neural Networks Bias And Weights. *FinTechExplained*. https://medium.com/fintechexplained/neural-networks-bias-and-weights-10b53e6285da

Marie, B. (2023, July 18). *QLoRa: Fine-Tune a Large Language Model on Your GPU*. Medium. https://towardsdatascience.com/qlora-fine-tune-a-large-language-model-on-your-gpu-27bed5a03e2b

Mathieu, J. (Ed.). (2022). *Contribution of storage technologies to renewable energy hubs*.

Maximum characters count in an Excel cell | XlsIO | Syncfusion. (n.d.). Retrieved July 24, 2023, from https://help.syncfusion.com/file-formats/xlsio/faqs/information-about-maximum-characters-count-in-an-excel-cell

Med-PaLM. (n.d.). Med-PaLM. Retrieved July 23, 2023, from https://sites.research.google/med-palm/

Meep. (2019, May 22). *What are tokens and tokenizations?* [Forum post]. Data Science Stack Exchange. https://datascience.stackexchange.com/q/52367

Memory | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://python.langchain.com/docs/modules/memory/

Meta's powerful AI language model has leaked online—What happens now? - The Verge. (n.d.). Retrieved July 23, 2023, from https://www.theverge.com/2023/3/8/23629362/meta-ai-language-model-llama-leak-online-misuse

Models. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/main_classes/model

Models—Hugging Face. (2023, March 9). https://huggingface.co/models

Modules | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://python.langchain.com/docs/modules/

Moret, S. (2017). *Strategic energy planning under uncertainty* [EPFL]. https://doi.org/10.5075/epfl-thesis-7961

mrbullwinkle. (2023, July 21). *Azure OpenAI Service models—Azure OpenAI*. https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models

Nader Dabit (Director). (2023, March 24). *Fine-tuning GPT from Scratch in 15 Minutes with JavaScript*. https://www.youtube.com/watch?v=Sb7U32kXMB0

Namazifar, M., Tur, G., & Hakkani-Tür, D. (2021). Warped Language Models for Noise Robust Language Understanding. *2021 IEEE Spoken Language Technology Workshop (SLT)*, 981–988. https://doi.org/10.1109/SLT48900.2021.9383493

NBA Players Performance. (n.d.). Retrieved July 23, 2023, from https://www.kaggle.com/datasets/thedevastator/unlocking-the-secrets-of-nba-player-performance

Neural Networks basics | OpenNN Start. (n.d.). Retrieved July 23, 2023, from https://www.opennn.net/documentation/neural_networks_basics.html

New and improved embedding model. (n.d.). Retrieved July 24, 2023, from https://openai.com/blog/new-and-improved-embedding-model

Nicholas Renotte (Director). (2020). *Tensorflow Tutorial for Python in 10 Minutes*. https://www.youtube.com/watch?v=6_2hzRopPbQ

Nicholls, J. (2018, August 13). Quantization in Deep Learning. *Medium*. https://medium.com/@joel_34050/quantization-in-deep-learning-478417eab72b

Nordhaus, W. D. (2017). Revisiting the social cost of carbon. *Proceedings of the National Academy of Sciences*, *114*(7), 1518–1523. https://doi.org/10.1073/pnas.1609244114

oaclaf. (n.d.). *uuid0: A library to make better timestamped UUIDs for databases and web apps* (0.2.7) [Python; OS Independent]. Retrieved July 24, 2023, from https://github.com/oaclaf/uuid0

On word embeddings—Part 1. (2016, April 11). Ruder.Io. https://www.ruder.io/word-embeddings-1/

OpenAI. (n.d.). Retrieved May 14, 2023, from https://openai.com/

OpenAI | 🦜🔗 Langchain. (n.d.). Retrieved July 23, 2023, from https://python.langchain.com/docs/modules/data_connection/text_embedding/integrations/openai

OpenAI API. (n.d.). Retrieved July 24, 2023, from https://openai.com/blog/openai-api

OpenAI GPT. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/model_doc/openai-gpt

OpenAI Platform. (n.d.). Retrieved July 24, 2023, from https://platform.openai.com

OpenAI Quietly Released GPT-3.5: Here's What You Can Do With It | by Clément Bourcart | DataDrivenInvestor. (n.d.). Retrieved July 23, 2023, from https://medium.datadriveninvestor.com/openai-quietly-released-gpt-3-5-heres-what-you-can-do-with-it-4dee22aea438

OpenAIEmbeddings | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://js.langchain.com/docs/api/embeddings_openai/classes/OpenAIEmbeddings

Open-Llama. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/main/model_doc/open-llama

OpenLLaMA: An Open Reproduction of LLaMA. (2023a). [Computer software]. OpenLM Research. https://github.com/openlm-research/open_llama (Original work published 2023)

OpenLLaMA: An Open Reproduction of LLaMA. (2023b). [Computer software]. OpenLM Research. https://github.com/openlm-research/open_llama (Original work published 2023)

OpenLLaMa—An Open Reproduction of LLaMA | MLExpert—Crush Your Machine Learning interview.

(n.d.). Retrieved July 23, 2023, from https://mlexpert.io/prompt-engineering/open-llama

Overcoming AI's Limitations to Reach True Understanding. (2022, July 6). Devmio - Software Know-How. https://devm.io/machine-learning/ai-limitations-data-177594-001

Overview. (n.d.). Pinecone. Retrieved July 24, 2023, from https://docs.pinecone.io/docs

Özpoyraz, İ. (2022, December 16). The Limitations of AI: Why Generalization is a Challenge. *KoçDigital*. https://medium.com/kocdigital/the-limitations-of-ai-why-generalization-is-a-challenge-59c41e78a655

PaLM API | Generative AI for Developers. (n.d.). Retrieved May 22, 2023, from https://developers.generativeai.google/products/palm

Pandas | Python Library—Mode. (2016, May 23). Mode Resources. https://mode.com/python-tutorial/libraries/pandas/

Pandas Tutorial. (n.d.). Retrieved July 23, 2023, from https://www.w3schools.com/python/pandas/default.asp

pandas—Python Data Analysis Library. (n.d.). Retrieved July 24, 2023, from https://pandas.pydata.org/

Parthasarathy, S. (2023, April 3). The Significance of Vicuna, an Open-Source Large Language Model for Chatbots. *MLearning.Ai*. https://medium.com/mlearning-ai/the-significance-of-vicuna-an-open-source-large-language-model-for-chatbots-23b4765711ff

PDF | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://python.langchain.com/docs/modules/data_connection/document_loaders/pdf

PEFT. (n.d.-a). Retrieved July 23, 2023, from https://huggingface.co/docs/peft/index

PEFT. (n.d.-b). Retrieved July 23, 2023, from https://huggingface.co/docs/peft/index

Performance and Scalability: How To Fit a Bigger Model and Train It Faster. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/v4.18.0/en/performance

Pietsch, M., Möller, T., Kostic, B., Risch, J., Pippi, M., Jobanputra, M., Zanzottera, S., Cerza, S., Blagojevic, V., Stadelmann, T., Soni, T., & Lee, S. (2019). *Haystack: The end-to-end NLP framework for pragmatic builders* [Python]. https://github.com/deepset-ai/haystack (Original work published 2019)

Pinecone. (n.d.). Retrieved July 24, 2023, from https://docs.pinecone.io/

Pinecone is now available on the Google Cloud Marketplace | Pinecone. (n.d.). Retrieved July 24, 2023, from https://www.pinecone.io/blog/pinecone-gcp-marketplace/

pinecone with open source embeding model—Google Search. (n.d.). Retrieved July 19, 2023, from https://www.google.com/search?q=pinecone+with+open+source+embeding+model&rlz=1C1VDKB_deCH1045CH1045&oq=pinecone+with+open+source+embeding+model&aqs=chrome..69i57j33i10i160l2.9068j0j7&sourceid=chrome&ie=UTF-8

Pipeline Components Overview. (n.d.). Haystack Documentation. Retrieved July 24, 2023, from https://docs.haystack.deepset.ai/docs

Politi, M. (2023, June 22). 🦜🔗*LangChain: Enhancing Performance with Memory Capacity*. Medium. https://towardsdatascience.com/langchain-enhancing-performance-with-memory-capacity-c7168e097f81

Possible to share finetuned model with another user? - API. (2021, July 26). OpenAI Developer Forum. https://community.openai.com/t/possible-to-share-finetuned-model-with-another-user/6743

Pradip Nichite (Director). (2023). *Semantic Search with Open-Source Vector DB: Chroma DB | Pinecone Alternative | Code*. https://www.youtube.com/watch?v=kCL4JuRPD6U

Pramoditha, R. (2022, October 31). Deep Learning Hardware Selection Guide for 2023. *Data Science 365*. https://medium.com/data-science-365/deep-learning-hardware-selection-guide-for-2023-601808ee3a24

Pretrained Models—Sentence-Transformers documentation. (n.d.). Retrieved July 24, 2023, from https://www.sbert.net/docs/pretrained_models.html

Pricing | Pinecone. (n.d.). Retrieved July 19, 2023, from https://www.pinecone.io/pricing/

Privacy policy. (n.d.). Retrieved July 24, 2023, from https://openai.com/policies/privacy-policy

Product | Pinecone. (n.d.). Retrieved July 24, 2023, from https://www.pinecone.io/product/

Python Client. (n.d.). Pinecone. Retrieved July 24, 2023, from https://docs.pinecone.io/docs/python-client

Python Simplified (Director). (2021a). *Machine Learning FOR BEGINNERS - Supervised, Unsupervised and Reinforcement Learning*. https://www.youtube.com/watch?v=mMc_PIemSnU

Python Simplified (Director). (2021b). *Perceptron Algorithm with Code Example—ML for beginners!* https://www.youtube.com/watch?v=-KLnurhX-Pg

Python Simplified (Director). (2021c, October 16). *Neural Network Simply Explained—ML for Beginners*. https://www.youtube.com/watch?v=i1AqHG4k8mE

PyTorch. (n.d.). Retrieved July 19, 2023, from https://pytorch.org/

PyTorch: How to Set .requires_grad False | Saturn Cloud Blog. (2023, July 10). https://saturncloud.io/blog/pytorch-how-to-set-requiresgrad-false/

QA and Chat over Documents | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://python.langchain.com/docs/use_cases/question_answering/

Quantization. (n.d.). Retrieved July 22, 2023, from https://huggingface.co/docs/optimum/concept_guides/quantization

Quickstart. (n.d.-a). Pinecone. Retrieved July 24, 2023, from https://docs.pinecone.io/docs

Quickstart. (n.d.-b). Pinecone. Retrieved July 24, 2023, from https://docs.pinecone.io/docs/quickstart

Raffaele. (2021, May 5). *Answer to "In the context of Deep Learning, what is training warmup steps."* Data Science Stack Exchange. https://datascience.stackexchange.com/a/94013

Raizada, A. (2023, March 29). The Limitations of AI & Machine Learning. *Copper Digital*. https://copperdigital.com/blog/the-limitations-of-ai-and-machine-learning/

Reader. (n.d.). Haystack Documentation. Retrieved July 24, 2023, from https://docs.haystack.deepset.ai/docs/document_store

Reading Text from the Image using Tesseract—GeeksforGeeks. (n.d.). Retrieved July 24, 2023, from https://www.geeksforgeeks.org/reading-text-from-the-image-using-tesseract/

RecursiveCharacterTextSplitter | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://js.langchain.com/docs/api/text_splitter/classes/RecursiveCharacterTextSplitter

Red Pajama Is a 1.2 Trillion Token Large Language Model | NextBigFuture.com. (2023, April 19). https://www.nextbigfuture.com/2023/04/red-pajama-is-a-1-2-trillion-token-large-language-model.html

RedPajama, a project to create leading open-source models, starts by reproducing LLaMA training dataset of over 1.2 trillion tokens. (n.d.). TOGETHER. Retrieved May 30, 2023, from https://www.together.xyz/blog/redpajama

RedPajama replicates LLaMA dataset to build open source, state-of-the-art LLMs. (2023, April 18). *VentureBeat*. https://venturebeat.com/ai/redpajama-replicates-llama-to-build-open-source-state-of-the-art-llms/

RedPajama-Data: An Open Source Recipe to Reproduce LLaMA training dataset. (2023). [Python]. Together. https://github.com/togethercomputer/RedPajama-Data (Original work published 2023)

Rehberg, J. (2022, October 24). *High scalable fast search with Haystack*. Medium. https://towardsdatascience.com/high-scalable-fast-search-with-haystack-8b7bb103df8e

Reinforcement learning. (2023). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Reinforcement_learning&oldid=1155454334

Rent GPUs | Vast.ai. (n.d.). Vast AI. Retrieved July 24, 2023, from https://vast.ai/

Retriever. (n.d.-a). Haystack Documentation. Retrieved July 24, 2023, from https://docs.haystack.deepset.ai/docs/retriever

Retriever. (n.d.-b). Haystack Documentation. Retrieved July 24, 2023, from https://docs.haystack.deepset.ai/docs/document_store

Risks and limitations of artificial intelligence in business | nibusinessinfo.co.uk. (n.d.). Retrieved May 30, 2023, from https://www.nibusinessinfo.co.uk/content/risks-and-limitations-artificial-intelligence-business

roganjosh. (2020, January 16). *Answer to “Spyder 4 is not displaying plots and displays message like this ‘uncheck “Mute Inline Plotting” under the Plots pane options menu.’”* Stack Overflow. https://stackoverflow.com/a/59769720

Safety & responsibility. (n.d.). Retrieved July 24, 2023, from https://openai.com/safety

Sagar, R. (2019, May 25). *What Does Freezing A Layer Mean And How Does It Help In Fine Tuning N* *eural Networks*. Analytics India Magazine. https://analyticsindiamag.com/what-does-freezing-a-layer-mean-and-how-does-it-help-in-fine-tuning-neural-networks/

Sam Witteveen (Director). (2023a). *Fine-tuning LLMs with PEFT and LoRA*. https://www.youtube.com/watch?v=Us5ZFp16PaU

Sam Witteveen (Director). (2023b). *LangChain Retrieval QA Over Multiple Files with ChromaDB*. https://www.youtube.com/watch?v=3yPBVii7Ct0

Sangmini/msmarco-cotmae-MiniLM-L12_en-ko-ja · Hugging Face. (n.d.). Retrieved July 24, 2023, from https://huggingface.co/sangmini/msmarco-cotmae-MiniLM-L12_en-ko-ja

Schnidrig, J. (Ed.). (2018). *EnergyScope Valais—Case study of Regionalisation*.

Schnidrig, J. (Ed.). (2020). *Assessment of green mobility scenarios on European energy systems*.

Schnidrig, J., Brun, J., Maréchal, F., & Margni, M. (Eds.). (2023). Integration of Life Cycle Impact Assessment in Energy System Modelling. *Proceedings of ECOS 2023*.

Schnidrig, J., Cherkaoui, R., Calisesi, Y., Margni, M., & Maréchal, F. (Eds.). (2023). On the role of energy infrastructure in the energy transition. Case study of an energy independent and CO2 neutral energy system for Switzerland. *Frontiers in Energy Research*. https://doi.org/10.3389/fenrg.2023.1164813

Schnidrig, J., Li, X., & Maréchal, F. (Eds.). (2022). Assessment of the role of infrastructure in high share renewable energy systems. *Proceedings of ECOS 2022*.

Schnidrig, J., Nguyen, T.-V., Li, X., & Maréchal, F. (Eds.). (2021a). A modelling framework for assessing the impact of green mobility technologies on energy systems. *Proceedings of ECOS 2021*.

Schnidrig, J., Nguyen, T.-V., Li, X., & Maréchal, F. (Eds.). (2021b). Appendix—A modelling framework for assessing the impact of green mobility technologies on energy systems. *Proceedings of ECOS 2021 - The 34th International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems*.

Schreiner, M. (2022, April 23). *Deepmind Chinchilla: Artificial Intelligence is far from being fed up*. THE DECODER. https://the-decoder.com/deepmind-artificial-intelligence-is-far-from-being-fed-up/

Schreiner, M. (2023, May 5). *OpenLLaMA is a fully open-source LLM, now ready for business*. THE

DECODER. https://the-decoder.com/openllama-is-a-fully-open-source-llm-now-ready-for-business/

Schwessinger, R. (2019, July 19). *Answer to "In the context of Deep Learning, what is training warmup steps."* Data Science Stack Exchange. https://datascience.stackexchange.com/a/56005

Scikit Learn Tutorial. (n.d.). Retrieved July 23, 2023, from https://www.tutorialspoint.com/scikit_learn/index.htm

Scikit-learn: Machine learning in Python—Scikit-learn 1.3.0 documentation. (n.d.). Retrieved July 23, 2023, from https://scikit-learn.org/stable/

Security | Pinecone. (n.d.). Retrieved July 24, 2023, from https://www.pinecone.io/security/

Seldon. (2022, September 16). *Supervised vs Unsupervised Learning Explained*. Seldon. https://www.seldon.io/supervised-vs-unsupervised-learning-explained

SentenceTransformers Documentation—Sentence-Transformers documentation. (n.d.). Retrieved July 23, 2023, from https://www.sbert.net/

Sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 · Hugging Face. (n.d.). Retrieved July 19, 2023, from https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

Sequential—PyTorch 2.0 documentation. (n.d.). Retrieved July 23, 2023, from https://pytorch.org/docs/stable/generated/torch.nn.Sequential.html

Session State—Streamlit Docs. (n.d.). Retrieved July 24, 2023, from https://docs.streamlit.io/

Setup—Auto-GPT. (n.d.). Retrieved July 19, 2023, from https://docs.agpt.co/setup/

Shenwai, T. (2023, May 4). Finetuning LLaMA on Medical Papers: Meet PMC-LLaMA-A Model that Achieves High Performance on Biomedical QA Benchmarks. *MarkTechPost*. https://www.marktechpost.com/2023/05/04/finetuning-llama-on-medical-papers-meet-pmc-llama-a-model-that-achieves-high-performance-on-biomedical-qa-benchmarks/

Simone. (2020, December 10). *Answer to "In the context of Deep Learning, what is training warmup steps."* Data Science Stack Exchange. https://datascience.stackexchange.com/a/86499

Singh, N. (2023, May 5). Meet OpenLLaMA: An Open-Source Reproduction of Meta AI's LLaMA Large

Language Model. *MarkTechPost*. https://www.marktechpost.com/2023/05/05/meet-openllama-an-open-source-reproduction-of-meta-ais-llama-large-language-model/

Slaymaker, A. (Ed.). (2021). *Demographic and Geographic Region Definition in Energy System Modelling. A case study of Canada's path to net zero greenhouse gas emissions by 2050 and the role of hydrogen*.

Souttre, M. (Ed.). (2022). *Prospective study on the cost evolution for key energy technologies*.

Spyder 4 is not displaying plots and displays message like this "uncheck 'Mute Inline Plotting' under the Plots pane options menu." (2020, January 16). [Forum post]. Stack Overflow. https://stackoverflow.com/q/59769492

Stanford Alpaca: An Instruction-following LLaMA Model. (2023). [Python]. Tatsu's shared repositories. https://github.com/tatsu-lab/stanford_alpaca (Original work published 2023)

Stanford CRFM. (n.d.). Retrieved May 21, 2023, from https://crfm.stanford.edu/2023/03/13/alpaca.html

StatQuest with Josh Starmer (Director). (2018, September 17). *Machine Learning Fundamentals: Bias and Variance*. https://www.youtube.com/watch?v=EuBBz3bI-aA

Streamlit Docs. (n.d.). Retrieved July 24, 2023, from https://docs.streamlit.io/library/api-reference/widgets/st.checkbox

Streefkerk, R. (2019, December 13). *Citer ses sources dans le texte avec les normes APA*. Scribbr. https://www.scribbr.fr/normes-apa/citer-des-sources-dans-le-texte-avec-apa/

SU, H. (n.d.). *InstructorEmbedding: Text embedding tool* (1.0.1) [Computer software]. Retrieved July 19, 2023, from https://github.com/HKUNLP/instructor-embedding

Summary of the tokenizers. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/docs/transformers/tokenizer_summary

Supervised Machine learning—Javatpoint. (n.d.). Retrieved July 22, 2023, from https://www.javatpoint.com/supervised-machine-learning

Takyar, A. (2023a, May 16). *A guide to Parameter-efficient Fine-tuning(PEFT)*. LeewayHertz - AI Development Company. https://www.leewayhertz.com/parameter-efficient-fine-tuning/

Takyar, A. (2023b, May 29). *A comprehensive guide to prompt engineering*. LeewayHertz - AI Development Company. https://www.leewayhertz.com/prompt-engineering/

Tam, A. (2023, May 28). A Gentle Introduction to Prompt Engineering. *MachineLearningMastery.Com*. https://machinelearningmastery.com/a-gentle-introduction-to-prompt-engineering/

Team, T. A. E. (2022, September 30). *Best Public Datasets for Machine Learning and Data Science*. Medium. https://pub.towardsai.net/best-datasets-for-machine-learning-data-science-computer-vision-nlp-ai-c9541058cf4f

TensorFlow API Versions | TensorFlow v2.13.0. (n.d.). Retrieved July 23, 2023, from https://www.tensorflow.org/versions

Terms & policies. (n.d.). Retrieved July 24, 2023, from https://openai.com/policies

Terrasi, V. (2023, March 22). *GPT-4: How Is It Different From GPT-3.5?* Search Engine Journal. https://www.searchenginejournal.com/gpt-4-vs-gpt-3-5/482463/

Tesseract OCR. (2023). [C++]. tesseract-ocr. https://github.com/tesseract-ocr/tesseract (Original work published 2014)

Testing p2 Pods, Vertical Scaling, and Collections | Pinecone. (n.d.). Retrieved July 24, 2023, from https://www.pinecone.io/learn/testing-p2-collections-scaling/

Text Classification: What it is And Why it Matters. (n.d.). MonkeyLearn. Retrieved July 23, 2023, from https://monkeylearn.com/text-classification/

Text embedding models | 🦜🔗 Langchain. (n.d.). Retrieved July 23, 2023, from https://python.langchain.com/docs/modules/data_connection/text_embedding/

Text Embeddings Visually Explained. (2022, June 28). Context by Cohere. https://txt.cohere.com/text-embeddings/

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models* (arXiv:2104.08663). arXiv. https://doi.org/10.48550/arXiv.2104.08663

thanatoz. (2019, May 22). *Answer to "What are tokens and tokenizations?"* Data Science Stack Exchange. https://datascience.stackexchange.com/a/52373

The AI Arms Race Is On. Start Worrying. (2023, February 16). Time. https://time.com/6255952/ai-impact-chatgpt-microsoft-google/

The AI-native open-source embedding database. (n.d.). Retrieved July 23, 2023, from https://www.trychroma.com

The Big 5 AI problems. Surely you must have heard how AI could… | by Srini Janarthanam | InfiniteThoughts | Medium. (n.d.). Retrieved May 30, 2023, from https://medium.com/infinitethoughts/the-big-5-ai-problems-f625adf299a

The Full Guide to Training Datasets for Machine Learning. (n.d.). Retrieved May 30, 2023, from https://encord.com/blog/an-introduction-to-data-labelling-and-training-data/

The Impact of Artificial Intelligence on 5 Industries. (n.d.). Retrieved July 22, 2023, from https://www.hotjar.com/blog/ai-impact-industries-1/

The Ultimate Guide to Deep Learning Model Quantization and Quantization-Aware Training. (n.d.). Deci. Retrieved July 22, 2023, from https://deci.ai/quantization-and-quantization-aware-training/

Thissen, M. (2023, March 27). How To Fine-Tune the Alpaca Model For Any Language | ChatGPT Alternative. *Medium*. https://medium.com/@martin-thissen/how-to-fine-tune-the-alpaca-model-for-any-language-chatgpt-alternative-370f63753f94

Top 5 Problems With AI That Remain Unsolved—BairesDev. (2022, May 31). BairesDev Blog: Insights on Software Development & Tech Talent. https://www.bairesdev.com/blog/problems-with-ai-that-remain-unsolved/

Tqdm · PyPI. (n.d.). Retrieved July 24, 2023, from https://pypi.org/project/tqdm/

Train With Mixed Precision. (n.d.). NVIDIA Docs. Retrieved July 23, 2023, from https://docs.nvidia.com/deeplearning/performance/mixed-precision-training/index.html

Trainer—Transformers 3.0.2 documentation. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/transformers/v3.0.2/main_classes/trainer.html

Training a new tokenizer from an old one—Hugging Face NLP Course. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/learn/nlp-course/chapter6/2

Troubleshooting vector dimension does not match the dimension of the index—Support. (2023, July 6). Pinecone Community. https://community.pinecone.io/t/troubleshooting-vector-dimension-does-not-match-the-dimension-of-the-index/978/7

Try Bard, an AI experiment by Google. (n.d.). Retrieved May 30, 2023, from https://bard.google.com

Tsjolder, M. (2020, April 9). *Answer to "In the context of Deep Learning, what is training warmup steps."* Data Science Stack Exchange. https://datascience.stackexchange.com/a/72039

Underscore_ (Director). (2023, March 23). *Pourquoi ChatGPT vient de se faire détrôner*. https://www.youtube.com/watch?v=uvnZOjPeV6M

Understand Classification Performance Metrics | by Alex Guanga | Becoming Human: Artificial Intelligence Magazine. (n.d.). Retrieved July 24, 2023, from https://becominghuman.ai/understand-classification-performance-metrics-cad56f2da3aa?gi=a45bb6a30670

Understanding LSTM Networks—Colah's blog. (n.d.). Retrieved May 31, 2023, from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Unsupervised learning. (2023). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Unsupervised_learning&oldid=1155869354#Training

Use Bard—Android—Bard Help. (n.d.). Retrieved July 23, 2023, from https://support.google.com/bard/answer/13275745?hl=en&co=GENIE.Platform%3DAndroid

Using LoRA for Efficient Stable Diffusion Fine-Tuning. (n.d.). Retrieved July 23, 2023, from https://huggingface.co/blog/lora

Vector Database for Vector Search | Pinecone. (n.d.). Retrieved July 19, 2023, from https://www.pinecone.io/

Vector Databases as Memory for your AI Agents | by Ivan Campos | Sopmac AI | Medium. (n.d.). Retrieved July 24, 2023, from https://medium.com/sopmac-ai/vector-databases-as-memory-for-your-ai-agents-986288530443

Vector search just got up to 10x faster, easier to set up, and vertically scalable | Pinecone. (n.d.). Retrieved July 24, 2023, from https://www.pinecone.io/blog/faster-easier-scalable/

Vector stores | 🦜🔗 Langchain. (n.d.). Retrieved July 24, 2023, from https://python.langchain.com/docs/integrations/vectorstores/

Venelin Valkov (Director). (2023, March 31). *Fine-tuning Alpaca: Train Alpaca LoRa for Sentiment Analysis on a Custom Dataset*. https://www.youtube.com/watch?v=4-Q50fmq7Uw

Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality | LMSYS Org*. (n.d.). Retrieved May 30, 2023, from https://lmsys.org/blog/2023-03-30-vicuna

Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality | LMSYS Org*. (n.d.). Retrieved July 23, 2023, from https://lmsys.org/blog/2023-03-30-vicuna/

Vicuna—The Unparalleled Open-Source AI Model for Local Computer Installation | by NapSaga | Artificial Intelligence in Plain English. (n.d.). Retrieved July 23, 2023, from https://ai.plainenglish.io/vicuna-the-unparalleled-open-source-ai-model-for-local-computer-installation-334c693c4931

Vincent, J. (2021, December 8). *DeepMind tests the limits of large AI language systems with 280-billion-parameter model*. The Verge. https://www.theverge.com/2021/12/8/22822199/large-language-models-ai-deepmind-scaling-gopher

Vincent, J. (2023a, March 8). *Meta's powerful AI language model has leaked online—What happens now?* The Verge. https://www.theverge.com/2023/3/8/23629362/meta-ai-language-model-llama-leak-online-misuse

Vincent, J. (2023b, March 8). *Meta's powerful AI language model has leaked online—What happens now?* The Verge. https://www.theverge.com/2023/3/8/23629362/meta-ai-language-model-llama-leak-online-misuse

Vyborov, E. (2023, February 20). GPT-3 API Latency—Model Comparison. *Medium*. https://medium.com/@evyborov/gpt-3-api-latency-model-comparison-13888a834938

Wang, E. J. (2023). 🦙🌲🤏 *Alpaca-LoRA* [Jupyter Notebook]. https://github.com/tloen/alpaca-lora (Original work published 2023)

Website Traffic—Check and Analyze Any Website. (n.d.). Similarweb. Retrieved July 24, 2023, from https://www.similarweb.com/

Weight (Artificial Neural Network). (2019, May 17). DeepAI. https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network

Weiss, T. R. (2021, December 8). *DeepMind Experimenting with Its Nascent Gopher 280 Billion Parameter Language Model*. EnterpriseAI.

https://www.enterpriseai.news/2021/12/08/deepmind-experimenting-with-its-nascent-gopher-280-billion-parameter-language-model/

Welcome to pypdf—Pypdf 3.13.0 documentation. (n.d.). Retrieved July 24, 2023, from https://pypdf.readthedocs.io/en/stable/

What are AI Hallucinations and Why Are They a Problem? TechTarget. (n.d.-a). WhatIs.Com. Retrieved July 19, 2023, from https://www.techtarget.com/whatis/definition/AI-hallucination

What are AI Hallucinations and Why Are They a Problem? TechTarget. (n.d.-b). WhatIs.Com. Retrieved July 22, 2023, from https://www.techtarget.com/whatis/definition/AI-hallucination

What are AI hallucinations—and how do you prevent them? | Zapier. (n.d.). Retrieved May 30, 2023, from https://zapier.com/blog/ai-hallucinations/

What are AI Tokens? A Guide For Beginners. (n.d.). Retrieved July 22, 2023, from https://sensoriumxr.com/articles/sensoriumxr.com/articles/what-are-ai-tokens

What Are Attention Masks? (2021, June 15). Luke Salamone's Blog. https://lukesalamone.github.io/posts/what-are-attention-masks/

What are foundation models? | IBM Research Blog. (n.d.). Retrieved July 22, 2023, from https://research.ibm.com/blog/what-are-foundation-models

What are tokens and how to count them? | OpenAI Help Center. (n.d.). Retrieved July 22, 2023, from https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them

What is a Vector Database? | Pinecone. (n.d.). Retrieved July 23, 2023, from https://www.pinecone.io/learn/vector-database/

What is Artificial Intelligence (AI)? | IBM. (n.d.). Retrieved July 22, 2023, from https://www.ibm.com/topics/artificial-intelligence

What is Artificial Intelligence and How Does AI Work? TechTarget. (n.d.). Enterprise AI. Retrieved July 22, 2023, from https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence

What is ChatGPT and why does it matter? Here's what you need to know | ZDNET. (n.d.). Retrieved July 23, 2023, from https://www.zdnet.com/article/what-is-chatgpt-and-why-does-it-matter-heres-everything-you-need-to-know/

What Is Embedding and What Can You Do with It | by Jinhang Jiang | Towards Data Science. (n.d.). Retrieved July 23, 2023, from https://towardsdatascience.com/what-is-embedding-and-what-can-you-do-with-it-61ba7c05efd8

What is Epoch in Machine Learning? | Simplilearn. (2022, August 30). Simplilearn.Com. https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learning

What is Generative AI? (n.d.). NVIDIA. Retrieved July 22, 2023, from https://www.nvidia.com/en-us/glossary/data-science/generative-ai/

What is Google Bard? Here's how to use this ChatGPT rival | Digital Trends. (n.d.). Retrieved July 23, 2023, from https://www.digitaltrends.com/computing/how-to-use-google-bard/

What is GPT-3? Everything You Need to Know - TechTarget. (n.d.). Enterprise AI. Retrieved July 23, 2023, from https://www.techtarget.com/searchenterpriseai/definition/GPT-3

What is GPT-4 and Why Does it Matter? (n.d.). Retrieved July 23, 2023, from https://www.datacamp.com/blog/what-we-know-gpt4

What is Gradient Accumulation in Deep Learning? | by Raz Rotenberg | Towards Data Science. (n.d.). Retrieved July 23, 2023, from https://towardsdatascience.com/what-is-gradient-accumulation-in-deep-learning-ec034122cfa

What is Haystack? (n.d.-a). Haystack. Retrieved July 19, 2023, from https://haystack.deepset.ai/overview/intro/

What is Keras and Why is it so Popular in 2023? (n.d.). Simplilearn.Com. Retrieved July 23, 2023, from https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras

What is Learning Rate in Machine Learning. (n.d.). Deepchecks. Retrieved July 23, 2023, from https://deepchecks.com/glossary/learning-rate-in-machine-learning/

What is low-rank adaptation (LoRA)? - TechTalks. (n.d.). Retrieved July 23, 2023, from https://bdtechtalks.com/2023/05/22/what-is-lora/

What is OpenAI? Definition and History from TechTarget. (n.d.). Retrieved July 23, 2023, from https://www.techtarget.com/searchenterpriseai/definition/OpenAI

What is Overfitting? - Overfitting in Machine Learning Explained - AWS. (n.d.). Amazon Web Services, Inc. Retrieved July 23, 2023, from https://aws.amazon.com/what-is/overfitting/

What is Prompt Engineering? – TechTarget Definition. (n.d.-a). Enterprise AI. Retrieved July 23, 2023, from https://www.techtarget.com/searchenterpriseai/definition/prompt-engineering

What is prompt engineering? Definition + skills | Zapier. (n.d.-b). Retrieved July 23, 2023, from https://zapier.com/blog/prompt-engineering/

What is Supervised Learning? | IBM. (n.d.). Retrieved July 22, 2023, from https://www.ibm.com/topics/supervised-learning

What is the minimum hardware requirement for trying out deep learning? (n.d.). Quora. Retrieved July 24, 2023, from https://www.quora.com/What-is-the-minimum-hardware-requirement-for-trying-out-deep-learning

What's ahead for Bard: More global, more visual, more integrated. (2023, May 10). Google. https://blog.google/technology/ai/google-bard-updates-io-2023/

Wimmer, P., Mehnert, J., & Condurache, A. P. (2023). Dimensionality Reduced Training by Pruning and Freezing Parts of a Deep Neural Network, a Survey. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-023-10489-1

Yes, Machines Make Mistakes: The 10 Biggest Flaws In Generative AI. (n.d.). Retrieved May 30, 2023, from https://uk.pcmag.com/news/146481/yes-machines-make-mistakes-the-10-biggest-flaws-in-generative-ai

Yu, F. (Joshua). (2023, May 8). Text Embedding—What, Why and How? *Medium*. https://medium.com/@yu-joshua/text-embedding-what-why-and-how-13227e983ba7

zchaarm. (2023, January 26). *R/LangChain Lounge* [Reddit Post]. R/LangChain. www.reddit.com/r/LangChain/comments/10ljho9/rlangchain_lounge/

Zulkifli, H. (2018, January 27). *Understanding Learning Rates and How It Improves Performance in Deep Learning*. Medium. https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10

🦜🔗 *LangChain alternatives*. (2023, June 14). Apify Blog. https://blog.apify.com/langchain-alternatives/

🧪 *Usage Guide | Chroma*. (n.d.). Retrieved July 23, 2023, from https://www.trychroma.com/usage-guide

Embeddings | Chroma. (n.d.). Retrieved July 23, 2023, from https://www.trychroma.com/embeddings

Annexes

Product backlog

User story	Role	Status	Priority
As an EPFL researcher, I want to have a detailed explanation on what is finetuning and how it can be implemented	EPFL researcher	Done	Must
As an EPFL researcher, I want to have examples of work or projects that implemented finetuning	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a comparison of the different libraries available	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a general overview of Artificial intelligence	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a detailed explanation on what is embedding and how it can be implemented	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a detailed explanation on the choice of framework used for the project	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a detailed explanation on the choice of embedding models used for the project	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a detailed explanation on the choice of AI used for the project	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a detailed explanation of the choice of vector database used for the project	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a detailed evaluation and explanation of the Q&A with sources implementation in the project	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a detailed evaluation and explanation of the scenario comparison implementation in the project	EPFL researcher	Done	Must
As an EPFL researcher, I want to have a detailed evaluation and explanation of the chat session implementation in the project	User	Done	Must

As a user, I want to be able to Q&A with the chatbot about Energyscope and energy related topics	User	Done	Must
As a user, I want to be able compare my scenario to other scenarios	User	Done	Must
As a user, I want to be able to keep a chat session with the AI	User	Done	Must
As a user, I want to be able to see the sources of information that the AI took to make up the answer	User	Done	Must
As a user, I want to be able to change the inputs of my scenario by asking the AI.	User	Not done	Optional
As a user, I want to be able to talk with the AI with speech to text.	User	Not done	Optional

Logbook

Weekly report: week 1

From 01.05.2023 to 07.05.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research	2	3	3	3	3	3	3	20
Development								
Writing	2							2

Report details

Day	Task
01.05.2023	Research about machine learning /writing
02.05.2023	Research about machine learning
03.05.2023	Research about machine learning
04.05.2023	Research about deep learning
05.05.2023	Research about deep learning
06.05.2023	Research about deep learning
07.05.2023	Research about artificial intelligence

Weekly report: week 2

From 08.05.2023 to 14.05.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research	3	3	3	3	3	3	3	21
Development								
Writing			3		2			5

Report details

Day	Task
08.05.2023	Product backlog creation
09.05.2023	Product backlog creation

10.05.2023	Following course on deep learning / writing
11.05.2023	Following course on deep learning
12.05.2023	Following course on deep learning / writing
13.05.2023	Following course on deep learning
14.05.2023	Following course on deep learning

Weekly report: week 3

From 15.05.2023 to 21.05.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research	3	3	3	3	3	3	3	21
Development								
Writing			3					3

Report details

Day	**Task**
15.05.2023	Searching the different AI on the market
16.05.2023	Searching the different AI on the market
17.05.2023	Searching the different AI on the market / writing
18.05.2023	Searching the different types of learning
19.05.2023	Searching the different types of learning
20.05.2023	Searching the different types of learning
21.05.2023	Searching the different types of models

Weekly report: week 4

From 22.05.2023 to 28.05.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research	3	3	3	3	3	3	3	21
Development								
Writing				3				3

Report details

Day	Task
22.05.2023	Research on LLaMA models
23.05.2023	Installation of LLaMA on my machine
24.05.2023	Finetuning of LLAMA
25.05.2023	Finetuning of LLAMA / writing
26.05.2023	Finetuning of LLAMA
27.05.2023	Finetuning of LLAMA
28.05.2023	Finetuning of LLAMA

Weekly report: week 5

From 23.05.2023 to 29.05.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research	3	3	3	3	3	3	3	21
Development								
Writing				3				3

Report details

Day	Task
23.05.2023	Research on OpenAI
24.05.2023	Finetuning a Curie model
25.05.2023	Finetuning a Curie model
26.05.2023	Finetuning a Curie model / writing
27.05.2023	Finetuning a Davinci model
28.05.2023	Finetuning a Davinci model
29.05.2023	Finetuning a Davinci model

Weekly report: week 6

From 30.05.2023 to 06.06.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research	3	6	8	8	7	7	8	41
Development								

Writing				4				4

Report details

Day	Task
30.05.2023	Research on AI deficiencies
01.06.2023	Research on AI deficiencies
02.06.2023	Research on Quantisation
03.06.2023	Research on Prompt engineering / writing
04.06.2023	Research on the different libraries available
05.06.2023	Research on the different libraries available
06.06.2023	Research on the different libraries available

Weekly report: week 7

From 06.06.2023 to 12.06.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research	7	7	8	4	7	7	5	45
Development								
Writing				4				4

Report details

Day	Task
06.06.2023	Research on PEFT and LoRA
07.06.2023	Research on PEFT and LoRA
08.06.2023	Research on PEFT and LoRA
09.06.2023	Research on Alpaca & Vicuna model / writing
10.06.2023	Research on Alpaca & Vicuna model
11.06.2023	OpenAI finetuning
12.06.2023	OpenAI finetuning

Weekly report: week 8

From 13.06.2023 to 19.06.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research	7	7	5	6	6	5	2	38
Development								
Writing			2					2

Report details

Day	Task
13.06.2023	Research on other alternatives than finetuning
14.06.2023	Research on other alternatives than finetuning
15.06.2023	Research on other alternatives than finetuning / writing
16.06.2023	Research on other alternatives than finetuning
17.06.2023	Research on other alternatives than finetuning
18.06.2023	Research on embedding
19.06.2023	Research on embedding

Weekly report: week 9

From 14.06.2023 to 20.06.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research	7	8	4	4	4	4	7	38
Development								
Writing			4	4	4	4		16

Report details

Day	Task
14.06.2023	Research and testing Langchain
15.06.2023	Research and testing Langchain
16.06.2023	Research and testing Langchain / writing
17.06.2023	Research and testing Langchain / writing
18.06.2023	Research and testing Pinecone / writing
19.06.2023	Research and testing Pinecone / writing
20.06.2023	Research and testing Langchain

Weekly report: week 10

From 21.06.2023 to 27.06.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research								
Development	8	7	4	0	5	7	8	39
Writing			4	6				10

Report details

Day	Task
21.06.2023	Linking of Pinecone with Langchain
22.06.2023	Linking of Pinecone with Langchain
23.06.2023	Linking of Pinecone with Langchain / writing
24.06.2023	Linking of Pinecone with Langchain / writing
25.06.2023	Implementation of Q&A
26.06.2023	Implementation of Q&A
27.06.2023	Implementation of Q&A

Weekly report: week 11

From 28.06.2023 to 04.07.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research					2			2
Development	8	7	6	6	4			31
Writing						8	8	16

Report details

Day	Task
28.06.2023	Data gathering
29.06.2023	Data gathering
30.06.2023	Creation of the dataset
01.07.2023	Creation of the dataset
02.07.2023	Creation of the dataset / writing
03.07.2023	Creation of the dataset / writing

04.07.2023	Testing Q&A with the dataset

Weekly report: week 12

From 05.07.2023 to 11.07.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research								
Development	6	4	3	3	3	2	2	23
Writing	3	3	2	3	4	3	2	20

Report details

Day	Task
05.06.2023	Implementation of Q/A with sources / writing
06.06.2023	Implementation of Q/A with sources / writing
07.06.2023	Implementation of Q/A with sources / writing
08.07.2023	Implementation of Q/A with sources / writing
09.07.2023	Implementation of Q/A with sources / writing
10.07.2023	Implementation of Q/A with sources / writing
11.07.2023	Implementation of scenario comparison / writing

Weekly report: week 13

From 12.07.2023 to 18.07.2023

Task	*Monday*	*Tuesday*	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sunday*	*Total*
Research								
Development	3	3	3	3	3	3	2	20
Writing	3	3	3	3	3	3	2	20

Report details

Day	Task
12.06.2023	Implementation of scenario comparison / writing

13.06.2023	Implementation of scenario comparison / Writing
14.06.2023	Implementation of scenario comparison / writing
15.07.2023	Implementation of scenario comparison / writing
16.07.2023	Implementation of scenario comparison / writing
17.07.2023	Implementation of Chat session / writing
18.07.2023	Implementation of Chat session / writing

Weekly report: week 14

From 19.07.2023 to 25.07.2023

Task	*Wednesday*	*Thursday*	*Friday*	*Saturday*	*Sundaay*	*Monday*	*Tuesday*	*Total*
Research								
Development	9	9	9					27
Writing				10	10	10	10	40

Report details

Day	Task
19.06.2023	Implementation of Chat session / writing
20.06.2023	Implementation of Chat session / writing
21.06.2023	Implementation of Chat session / writing
22.07.2023	Writing
23.07.2023	Writing
24.07.2023	Writing
25.07.2023	Writing

Product owner meetings

Date	Subject	Participants
09.5.2023	First meeting, introduction to the Bachelor thesis	Michel Lopez
24.05.2023	Discussion on where to go, structure of the bachelor thesis	Michel Lopez
28.05.2023	Discussion on the first version of the bachelor thesis, advice on references and small corrections on writing	David Wannier
31.05.2023	Discussion about LLaMA and OpenAI, general overview of the bachelor thesis	Michel Lopez
02.06.2023	Discussion about finetuning and his impact, adjustments of the bachelor thesis	Michel Lopez
05.06.2023	Discussion about OpenAI and the finetuning approach, overview of the bachelor thesis	Michel Lopez
20.06.2023	Meeting to discuss the progress of the bachelor thesis, including revisions to the text and guidance on future directions.	Michel Lopez
03.07.2023	Conducting the initial testing for the proof of concept and seeking recommendations to improve the project	Michel Lopez, François Maréchal
12.07.2023	Testing the Q&A with sources with scenario comparison, making the final decision regarding the inclusion of chat sessions	Michel Lopez, François Maréchal

Total of hours

Week	Research	Development	Writing
1	20	0	2
2	21	0	5
3	21	0	3
4	21	0	3
5	21	0	4
6	41	0	4
7	45	0	4
8	38	0	2
9	38	0	16
10	0	39	10
11	2	31	16
12	0	23	20
13	0	20	20
14	0	27	40
Total per category	268	140	149
Total	557 hours		

Hes·so VALAIS WALLIS
Haute Ecole de Gestion
Hochschule für Wirtschaft

Bachelor's thesis presentation

HES-SO Valais			Information regarding Bachelor's thesis	FO.2.2.02.28.EC mob/30/10/2017
FEE	FIG	FTO		
	X			

Degree programme: **BUSINESS INFORMATION TECHNOLOGY**

Confidential ☐

Student **SURNAME:** **BINJOS** Name: Abdullah **Mobile:** **+41 79 615 38 32**	**Year** 2022
Submitted by: François Maréchal (Industrial Process and Energy Systems Engineering) EPFL **Professor:** Wannier David	**Language** ☒ French ☐ German ☒ English

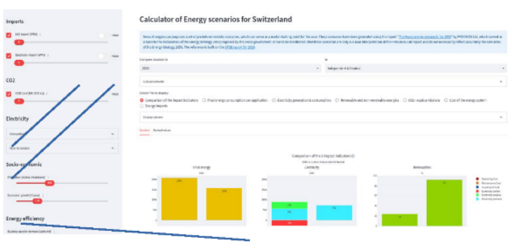
Title:

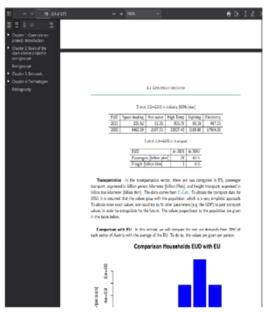
Intégration d'openAI pour l'exploration des solutions de la transition énergétique

Description:

L'objectif du projet est d'intégrer openai pour l'analyse des résultats de calculs de configuration de systèmes énergétique.

Les outils de l'IPESE (Industrial Process and Energy Systems Engineering) de l'EPFL génèrent des listes de configurations de systèmes énergétiques par l'utilisation d'algorithmes de calculs. Chaque configuration est définie par des caractéristiques techniques et des indicateurs de performances qui sont fournis sous la forme d'une table de résultats. Les colonnes définissent les caractéristiques et les lignes représentent les configurations. Pour chaque configuration, un rapport détaillé peut être généré. L'analyse des résultats permet ensuite d'identifier les meilleures configurations du système. Cela se fait par l'utilisation d'un système de cordonnées parallèles dans lequel l'utilisateur effectue la recherche de ses configurations préférentielles.
L'objectif du projet consiste à intégrer cette table dans l'outil openAI (Moteur utilisé par l'outil ChatGPT) pour aider à identifier les bonnes solutions en répondant aux interrogations d'un utilisateur qui cherche dans la table. Il conviendra d'apprendre au chatbot à lire la table de l'entrainer sur la manière de répondre aux questions et de récupérer les résultats de manière à aider l'utilisateur dans l'exploration des solutions.
L'utilisateur qui recevra comme réponse les configurations sélectionnées qui seront ensuite présentées comme sélections pour faire un rapport des solutions sous la forme de coordonnées parallèles et d'un rapport quarto.
Le travail consiste à développer les codes python ou R pour l'utilisation de OPENAI, intégration de fine tuning de l'OPENAI et une GUI de browsing et de présentation des résultats (utilisation de ggplot) et intégration dans l'outil de reporting quarto.





Hes·so VALAIS WALLIS
Haute Ecole de Gestion
Hochschule für Wirtschaft

HES-SO Valais		
FEE	FIG	FTO
	X	

Information regarding Bachelor's thesis

FO.2.2.02.28.EC
mob/30/10/2017

Planned project stages:

Analysis or state of the art

- Implémentation et/ou intégration d'openAI ou chatGPT dans une application tiers ;
- FineTunning des conversations avec l'IA pour l'apprentissage de la donnée ou du contexte de la conversation avec l'IA ;
- Compréhension et traduction de la réponse de l'IA pour la transférer vers un outil de reporting

Decision and justification by the student

Comme il s'agit d'un projet innovant, l'étudiant doit mettre un point d'honneur à la simplicité et à l'efficacité des différentes solutions trouvées. Le but ici est vraiment de trouver la solution optimale pour répondre aux besoins de ce projet.

Implementation & Testing

- Integration d'OpenAI / ChatGPT dans une application tiers : Tester la communication avec OpenAI
- FineTunning et setting de la communication dans un contexte distinct : Tester la cohérence des résultats donnés par l'IA
- Récupération des résultats et intégration de ces résultats dans un outil de reporting (texte, tableau ou graphique)

Available resources

Data

La data est une liste de configuration, Inputs et Outputs (résultats de calculs mathématiques) sous format JSON. La métric n'est pas un élément essentiel à l'heure actuelle car les calculs sont effectués en live, du au fait que les configurations possibles ont été évaluées à plus de 251 Yotta de possibilités.

Use case

Nous utiliserons ici un outil ENERGYSCOPE, crée par le laboratoire qui permet via une interface web de configurer une situation énergétique de la suisse et qui calcule une série de KPI comme résultat. L'idée est d'intégrer OpenAI dans cette interface pour le questionner dans un contexte d'analyse de ces résultats et de les récupérer pour les inclure dans un rapport.

Hardware & resources

Nous fournissons une machine virtuelle qui contient toutes les ressources et outils nécessaires à la mise en œuvre de ce projet.

Comments by the professor:

La comparaison des résultats obtenus par l'IA avec ceux obtenus avec d'autres outils comme ENERGYSCOPE sera effectué par l'équipe du professeur Marechal.

Signature	**Key dates**
Head degree programme Business Information Tech.	Start : **01.05.2023**
Professor:	Due date of report: **29.07.2023 12.00**
Student: A B	Public exhibition of Bachelor's theses: **22.09.2023**

Hes·so VALAIS WALLIS
Haute Ecole de Gestion
Hochschule für Wirtschaft

Author Declaration

I hereby declare, through this document, that I have carried out the attached Bachelor's work independently, without any help other than those duly mentioned in the references, and that I have used only the expressly mentioned sources. ChatGPT have been used to better understand which knowledge to know first to better understand the thesis. I will not provide a copy of this report to a third party without the joint permission of the RF and the professor in charge of supervising the Bachelor's work, including the applied research partner with whom I collaborated, except for the people who provided me with the main information necessary for writing this work and whom I cite below:

- David Wannier
- Michel Lopez
- François Maréchal

Date & location	Signature
26.07.2023	A B