



## GenAI resources (OBSOLETE)



In this section, we provide you with some technical guidelines about GenAI models and starting points on how to use them. For each data modality (text, image, audio, etc), we will provide you with the base models to consider, and some additional notebooks/resources that will help you getting started with these models.

### GenAI modalities

#### ▼ Text

This modality includes all natural text related tasks such as summarization, question answering, or language understanding in general. GenAI for text is made possible thanks to LLMs (Large Language Models) which are extensively parameterized deep neural network models trained for the task of next token prediction on large corpuses of data.

Next, we provide some guidelines on which LLMs to use and how to use them:

#### Base models

Base models are LLMs that have already been trained on large corpuses of data. They represent generalist models that can accomplish different text-related tasks. We suggest the following pre-trained LLMs from Hugging face's hub:

To make the right choice about which model to use, we suggest to look at the usual benchmarks used to rank LLMs:

- [Chatbot Arena](#)
- [Open LLM Leaderboard](#)
- <https://llm.extractum.io/>

N.B. These are **examples** of base models. It's part of your task to **check out the latest** ones and the **most adapted** to your case

Besides these general-purpose benchmarks, it might be judicious to choose the adequate base model **based on your specific usage**. For instance, some models might be designed for RAG applications, others might be more efficient for fine-tuning, etc. If needed, **get in touch with your mentors** for guidance in choosing the best LLM for your use-case.

Customizing Pre-trained LLMs can be achieved via:

- **Fine-tuning**
  - **In-context learning, or Retrieval Augmented Generation (RAG)**
- 

## Fine-tuning

Fine-tuning is a powerful training technique that involves training a pretrained model on a dataset that is specific to your particular task. By fine-tuning a pretrained model on a specific task, you can take advantage of the knowledge and expertise that the model has already acquired from its prior training on a vast and varied dataset, and tailor it to suit your unique requirements.

This can result in a significant boost in performance and accuracy, as compared to training a model from scratch.

- Basic steps to fine-tune a pretrained model:

Fine-tuning typically requires careful consideration of the quality of data that represent the specific task of interest.

Here are some useful resources for fine-tuning:

- [Fine-tuning a model with the Trainer API - Hugging Face NLP Course](#)
- Parameter-Efficient Fine-Tuning (PEFT): [blog](#) - [notebook\(Gemma-7B\)](#)
- Llama3: [Meta docs](#) - [Torchtune](#) - [ORPO](#)
- [Colab finetuning examples - Unsloth](#)

Additional resources on Parameter-efficient fine-tuning techniques for LLMs (such as LoRA):

- <https://github.com/huggingface/peft>
  - [https://github.com/peremartra/Large-Language-Model-Notebooks-Course/blob/main/5-Fine-Tuning/LoRA\\_Tuning\\_PEFT.ipynb](https://github.com/peremartra/Large-Language-Model-Notebooks-Course/blob/main/5-Fine-Tuning/LoRA_Tuning_PEFT.ipynb)
  - [https://colab.research.google.com/github/google/generative-ai-docs/blob/main/site/en/gemma/docs/lora\\_tuning.ipynb](https://colab.research.google.com/github/google/generative-ai-docs/blob/main/site/en/gemma/docs/lora_tuning.ipynb)
- 

## RAG

For LLMs, the pre-trained weights represent a sort of **parametric memory** that stores general knowledge, learned from large corpuses of training data. Fine-tuning aims at modifying this parametric memory to further store task-specific knowledge.

Another technique to incorporate task-tailored instructions into an LLM, is **Retrieval Augmented Generation (RAG)**. As opposed to fine-tuning, RAG aims at fetching data from another **non-parametric memory**, constituted of external

documents and knowledge. The LLM is therefore able to generate responses based on both, the general knowledge memory stored in its fixed weights, and the specific data retrieved using RAG.

In practice, RAG can be very useful for building a **chatbot interface with existing data**. For instance, a company can 'RAG' its internal documentation enabling its employees to 'chat' with it by means of an LLM.

We now provide some resources in knowing more and getting started with RAG:

- [Langchain chatgpt over your data - Repo](#)
- [Huggingface cookbook](#)
- [Survey repo](#)
- [Gemma RAG with Langchain notebook](#)
- [Simple local rag](#)
- [Llama3 RAG with Lightning AI](#)

## Prompt Engineering Resources

- [OpenAI guidelines](#)
- [OpenAI prompt examples](#)
- [Anthropic guidelines](#)
- [Anthropic prompt examples](#)

## ▼ Image

Similar to text, the image modality consist in models that can do **image generation** based on textual prompts. Popular models/tools in this area include: Dall-E, StableDiffusion, and Midjourney.

## Base models

| Base model                               | Link  | N params | Evaluation | Other |
|--|---|----------|------------|-------|
| Stable Diffusion v2-1                    | <a href="https://huggingface.co/stabilityai/stable-diffusion-2-1">https://huggingface.co/stabilityai/stable-diffusion-2-1</a>                 |          |            |       |
| stabilityai/stable-diffusion-xl-base-1.0 | <a href="https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0">https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0</a> |          |            |       |
| ByteDance/SDXL-Lightning                 | <a href="https://huggingface.co/ByteDance/SDXL-Lightning">https://huggingface.co/ByteDance/SDXL-Lightning</a>                                 |          |            |       |

## Diffusion models

Diffusion Models are a type of machine learning algorithms that is the most used for image generation. Furthermore, when fine-tuning these models, only a handful of images should be enough to get satisfactory results.

We now show some resources about these models and how to use them:

- [Projects from last year's hackathon](#)
- [Diffusers Training Examples](#)
- [Diffusion Models Course](#)
- [DreamBooth](#)
- Multi-Concept fine-tuning: [diffusers/examples/custom\\_diffusion at main · huggingface/diffusers · GitHub](#)
- [ControlNet](#)

## Prompt engineering

- Find the prompt from an image in [civitai](#) like in this [example](#)
  - <https://lexica.art/>

## ▼ Multimodal: Text + Image

Multimodal GenAI include models that can generate/understand **multiple modalities**. The most popular example are models that handle both text, and images.

This can be useful if we want to include context information in the form of images in text-to-image for instance. Currently the biggest models out there (GPT-4, Claude-3) are all multimodal, handling also different file formats such as pdf, or csv.

### Base models

| Base model                        | Link  | N params | Evaluation | Other |
|-----------------------------------|---|----------|------------|-------|
| llava-hf/llava-v1.6-mistral-7b-hf | <a href="https://huggingface.co/llava-hf/llava-v1.6-mistral-7b-hf">https://huggingface.co/llava-hf/llava-v1.6-mistral-7b-hf</a> |          |            |       |
| LLaVA-v1.6-34B                    | <a href="https://huggingface.co/liuhaotian/llava-v1.6-34b">https://huggingface.co/liuhaotian/llava-v1.6-34b</a>                 |          |            |       |
| idefics2-8b                       | <a href="https://huggingface.co/HuggingFaceM4/idefics2-8b">https://huggingface.co/HuggingFaceM4/idefics2-8b</a>                 |          |            |       |
| Paligemma                         | <a href="https://huggingface.co/blog/paligemma">https://huggingface.co/blog/paligemma</a>                                       |          |            |       |

Here is a [survey](#) about multimodal LLMs.

## ▼ Speech

The third modality that can be relevant in modern GenAI applications is **Speech**. This modality concerns all sort of **audio** data (soundwaves), with tasks such as: **Music generation**, **text-to-speech**, **Transcription**, and **Real-time Translation**.

We hereafter provide base models that can be used in voice-related GenAI applications:

### Base models

| Base model                      | Link  | N params | Evaluation | Other |
|---------------------------------|---|----------|------------|-------|
| Multilingual V2                 | <a href="https://huggingface.co/spaces/elevengirls/tts">https://huggingface.co/spaces/elevengirls/tts</a>                   |          |            |       |
| styletts2                       | <a href="https://huggingface.co/spaces/styletts2/styletts2">https://huggingface.co/spaces/styletts2/styletts2</a>           |          |            |       |
| parler-tts/parler_tts_mini_v0.1 | <a href="https://huggingface.co/parler-tts/parler_tts_mini_v0.1">https://huggingface.co/parler-tts/parler_tts_mini_v0.1</a> |          |            |       |
| distil-whisper/distil-large-v3  | <a href="https://huggingface.co/distil-whisper/distil-large-v3">https://huggingface.co/distil-whisper/distil-large-v3</a>   |          |            |       |

For a speech models leaderboard, check the following [link](#).

## ▼ Dynamical systems / Time series

**Dynamical systems** are functions which values **change over time**. Formally, the behavior of such systems can be described using (possibly stochastic) **differential equations**. Typical examples of such systems is the stock market prices and the epidemic evolution models, which are governed by popular models such as the Black & Scholes model.

The main interest in studying such systems is the task of **system identification**. This translates into learning the underlying dynamics of these systems from past data, and therefore, the ability to **predict their behavior** and take decisions accordingly (to invest or not based on a stock's price forecast).

When we dispose of long histories of past recordings and no assumptions about the governing equations of a dynamical system, we are rather faced with **Time series**. This machine learning field aims to study past recordings

of variables that change over time, potentially showing some **regularity** such as trends and seasonalities.

Time series machine learning models can generally be classified into two categories: Time series **forecasting** (e.g. weather forecasting), and time series **Classification** (e.g. earthquake detection from seismic movement recordings).

GenAI has been recently used to tackle both these problems, using LLMs and in-context learning:

- <https://github.com/ngruver/lilmtime>
- <https://github.com/AntonioLiu97/lilmICL>

Furthermore, we provide a list of recently-released time series foundational models:

- [Foundation Models for Time Series Analysis: A Tutorial and Survey](#)
- [Lag-llama](#)
- [Chronos: Learning the Language of Time Series](#)
- [Moment](#)

## ▼ Other (Video)

<https://huggingface.co/spaces/LanguageBind/Open-Sora-Plan-v1.0.0>

<https://huggingface.co/spaces/kadirnar/Open-Sora>

## Data sources

- Governmental open data

ACCUEIL | Portail de l'Open Data au Maroc

 <https://www.data.gov.ma/>

- Darija models and datasets from atlasia

atlasia (AtlasIA)

Open-source data and models for Morocco.

 <https://huggingface.co/atlasia>



 [huggingface.co/atlasia](https://huggingface.co/atlasia)

- Arabic models and datasets from arbml

ARBML

ARBML is a community of +700 researchers working on Arabic NLP research and development. Although we come from different walks of life, we all strive to achieve one goal: empowering our beloved language with open-source Arabic tools and applications.

<https://arbml.github.io/>

- Tafrigh - Converting multimedia data to arabic text

GitHub - ieasybooks/tafrigh: تفريغ المواد المرئية أو المسموعة إلى نصوص  
Contribute to ieasybooks/tafrigh development by creating an account on GitHub.

🔗 <https://github.com/ieeasybooks/tafrigh>



- MoroccoAI Arabic/Darija NLP resources

GitHub - MoroccoAI/Arabic-Darija-NLP-Resources: A curated collection of resources and repositories for Natural Language Processing (NLP) tasks specific to Darija, the Moroccan Arabic dialect. This repository

🔗 <https://github.com/MoroccoAI/Arabic-Darija-NLP-Resources>

## Prompt Engineering Resources

### Text

- [OpenAI guidelines](#)
- [OpenAI prompt examples](#)
- [Anthropic guidelines](#)
- [Anthropic prompt examples](#)

### Image

- Find the prompt from an image in [civital](#) like in this [example](#)
- <https://lexica.art/>