13-11-24

<https://drive.google.com/drive/folders/1VK5p_XwycvxdjuyA4r5q4mcWe3_p1rRS>

<https://github.com/shikharkumar13>

[Data Science Books - Google Drive](https://drive.google.com/drive/folders/1-1cvqxjY8OUZC87OLXcnusypgqXp4-bw)

[Generative AI - Scaler Topics](https://www.scaler.com/topics/generative-ai/)

[Generative AI Models - Scaler Topicshttps://en.wikipedia.org/wiki/Generative\_artificial\_intelligence](https://www.scaler.com/topics/artificial-intelligence-tutorial/generative-ai-models/)

[What is LSTM - Long Short Term Memory?](https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/)

[A black board with blue writing

Description automatically generated](https://www.scaler.com/topics/artificial-intelligence-tutorial/generative-ai-models/)

AF-Activation Function

[Long Short Term Memory Networks Explanation](https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/)

A diagram of a graph

Description automatically generated

14-11-24

[Gated Recurrent Unit Networks](https://www.geeksforgeeks.org/gated-recurrent-unit-networks/)

Transformer

[Transformers in Machine Learning](https://www.geeksforgeeks.org/getting-started-with-transformers/)

[How Transformers Work. Transformers are a type of neural](https://towardsdatascience.com/transformers-141e32e69591)

Attention Is All You Need

Retrieval-Augmented Generation

[RAG](https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/) (GPT=Transformer+RAG)

[Attention and Transformer Models](https://towardsdatascience.com/attention-and-transformer-models-fe667f958378)

[Self - attention in NLP](https://www.geeksforgeeks.org/self-attention-in-nlp/)

[Amazon SageMaker - AWS](https://www.googleadservices.com/pagead/aclk?sa=L&ai=DChcSEwj55MS49NqJAxXko4MHHXhmEmYYABAAGgJlZg&ae=2&aspm=1&co=1&ase=2&gclid=CjwKCAiAudG5BhAREiwAWMlSjMxFAQygHpE_AUBf0vMJZpfVwOcPKwVgzTDr1a2DN51pdoJoLWQ0eRoCNdMQAvD_BwE&ei=SHI1Z5z5BZ-Z4-EP7-uD2Aw&ohost=www.google.com&cid=CAESVuD2PNpaE6CZrrtnb8bc4iIlZj2bt-y91wW5nyM3tkj9NQ9R_YqmmI6yTklTuPQUAqog8SJios2lAjkvrqcxEjRflv6bOSHXgG-UxzeEdx-wBHDu48jl&sig=AOD64_2typ8yiRHdhlJ2yCJ0gMS7Fdhh-g&q&sqi=2&nis=6&adurl&ved=2ahUKEwicvbK49NqJAxWfzDgGHe_1AMsQ0Qx6BAgKEAE)

[Amazon Bedrock](https://aws.amazon.com/bedrock/?refid=fe192ccd-da4e-4c4a-9869-4c7b36c21981)

[Amazon Q](https://aws.amazon.com/q/)

[Azure AI Studio](https://azure.microsoft.com/en-in/products/ai-studio)

[Hugging Face Models](https://huggingface.co/models)

[Large Language Models (LLMs) with Google AI](https://cloud.google.com/ai/llms)

[Introduction to Large Language Models | Machine Learning](https://developers.google.com/machine-learning/resources/intro-llms)

A screenshot of a computer screen

Description automatically generated

A computer screen shot of a computer

Description automatically generated

15-11-24

Tranformer, BERT, GPT, LLM, Prompt

[Working\_with\_Huggingface.ipynb - Colab](https://colab.research.google.com/drive/1aOO88xoGC51I0Uspp9Z6xWFwxQgkbOMD#scrollTo=pKaugkS_1mZs)

<https://colab.research.google.com/drive/1aOO88xoGC51I0Uspp9Z6xWFwxQgkbOMD>

[GitHub - shikharkumar13/awesome-generative-ai-guide: A one stop repository for generative AI research updates, interview resources, notebooks and much more!](https://github.com/shikharkumar13/awesome-generative-ai-guide/tree/main)

<https://github.com/shikharkumar13/awesome-generative-ai-guide/tree/main>

<https://areganti.notion.site/Applied-LLMs-Mastery-2024-562ddaa27791463e9a1286199325045c>

<https://areganti.notion.site/Applied-LLMs-Mastery-2024-562ddaa27791463e9a1286199325045c>

[shikharkumar13 (shikharkumar13) / Repositories · GitHub](https://github.com/shikharkumar13?tab=repositories)

[GitHub - shikharkumar13/awesome-generative-ai-guide: A one stop repository for generative AI research updates, interview resources, notebooks and much more!](https://github.com/shikharkumar13/awesome-generative-ai-guide/tree/main)

[Applied LLMs Mastery 2024](https://areganti.notion.site/Applied-LLMs-Mastery-2024-562ddaa27791463e9a1286199325045c)

[Architecture and Working of Transformers in Deep Learning](https://www.geeksforgeeks.org/architecture-and-working-of-transformers-in-deep-learning/)

[Polo Club of Data Science, Georgia Tech](https://github.com/poloclub)

[Transformer Explained Visually: Learn How LLM ...](https://github.com/poloclub/transformer-explainer)

[Transformer Explainer: LLM Transformer Model Visually Explained](https://poloclub.github.io/transformer-explainer/)-POLO Club

[Models - Hugging Face](https://huggingface.co/models)

<https://www.youtube.com/@statquest>

<https://www.youtube.com/results?search_query=3Blue1Brown>

<https://daily.dev/>

[BERT Model – NLP](https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/)

<https://huggingface.co/docs/transformers/en/model_doc/bert>

[BERT-H2O](https://h2o.ai/wiki/bert/#:~:text=BERT%2C%20short%20for%20Bidirectional%20Encoder,framework%20for%20natural%20language%20processing.)

https://github.com/google-research/bert

[Understanding Q,K,V](https://medium.com/analytics-vidhya/understanding-q-k-v-in-transformer-self-attention-9a5eddaa5960)

[MLflow](https://mlflow.org/docs/latest/introduction/index.html)

A screenshot of a computer

Description automatically generated

12 self-attention,

A screenshot of a computer

Description automatically generated

MatMul-Matrice Multiplication

A screenshot of a computer code

Description automatically generated

A screenshot of a computer screen

Description automatically generated

19-11-24

[Vanishing and Exploding Gradients Problems in Deep](https://www.geeksforgeeks.org/vanishing-and-exploding-gradients-problems-in-deep-learning/)



[Vanishing Gradient Problem: Causes, Consequences, and ...](https://www.kdnuggets.com/2022/02/vanishing-gradient-problem.html)

21-11-24

LSTM Stock prediction

<https://colab.research.google.com/drive/13GLUshQfwodXpsAgw7k7gMRZG2JEnFpk#scrollTo=QT3SZkFZacpp>

model.add(LSTM(128, return\_sequences = True, input\_shape = (x\_train.shape[1], 1)))model.add(LSTM(64, return\_sequences = False))model.add(Dense(30))model.add(Dense(1))

<https://drive.google.com/drive/folders/1fPET7nRlKAVGt7sI8MSaWzSCZzWjYjgQ?usp=sharing>

[seq2seq Model in Machine Learning](https://www.geeksforgeeks.org/seq2seq-model-in-machine-learning/)

A diagram of a machine

Description automatically generated

[NLP — Vector Models](https://medium.com/thedeephub/nlp-vector-models-ca13f25aa450)

[Context Vectors](https://saturncloud.io/glossary/context-vectors/)

Resources for learning more about CoVe

To learn more about CoVe and its applications, you can explore the following resources:

* [Learned in Translation: Contextualized Word Vectors](https://arxiv.org/abs/1708.00107), the original paper introducing CoVe
* [Deep contextualized word representations](https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html#cove), a blog post discussing CoVe
* [Using pre-trained word embeddings in a Keras model](https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html), a tutorial on how to use pre-trained word embeddings, including CoVe, in a Keras model
* [Saturn Cloud](https://www.saturncloud.io/), a platform for free cloud compute resources to explore CoVe and other NLP techniques

[Saturn Cloud](https://app.community.saturnenterprise.io/dash/o/community/resources)-Important

[A Guide on Word Embeddings in NLP](https://www.turing.com/kb/guide-on-word-embeddings-in-nlp)

26-11-25

<https://drive.google.com/drive/folders/1fPET7nRlKAVGt7sI8MSaWzSCZzWjYjgQ?usp=sharing>

<https://colab.research.google.com/drive/1BgF_ZRW57BjvyUZEhhxEjibLWmF-ffN4?usp=sharing>

[Working with NLP](https://colab.research.google.com/drive/1aOO88xoGC51I0Uspp9Z6xWFwxQgkbOMD) through Hugging face

<https://14e00d32c34e56c0d7.gradio.live/>

<https://colab.research.google.com/drive/1BgF_ZRW57BjvyUZEhhxEjibLWmF-ffN4?usp=sharing>

https://buymeacoffee.com/dslim

[Named Entity Recognition](https://www.geeksforgeeks.org/named-entity-recognition/)

<https://dec6184c46ebabc65d.gradio.live/>

<https://ab1268b434511ea8a5.gradio.live>

<https://huggingface.co/dslim/bert-base-NER>

<https://www.gradio.app/>

[dslim/bert-base-NER · Hugging Face](https://huggingface.co/dslim/bert-base-NER)

<https://huggingface.co/dslim/bert-base-NER>

<https://drive.google.com/file/d/16hwWZUax_xCRU8f4Fw5VOWQ0lkmU59AJ/view?usp=sharing>

[Topic Modeling - Types, Working, Applications](https://www.geeksforgeeks.org/what-is-topic-modeling/)

[Topic Modeling - Types, Working, Applications](https://www.geeksforgeeks.org/what-is-topic-modeling/)

<https://colab.research.google.com/drive/1BgF_ZRW57BjvyUZEhhxEjibLWmF-ffN4?usp=sharing>

[Understanding TF-IDF (Term Frequency-Inverse ... https://drive.google.com/file/d/16hwWZUax\_xCRU8f4Fw5VOWQ0lkmU59AJ/view?usp=drive\_link](https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/)

<https://colab.research.google.com/drive/1BgF_ZRW57BjvyUZEhhxEjibLWmF-ffN4?usp=sharing>

27-11-24

<https://drive.google.com/drive/folders/1fPET7nRlKAVGt7sI8MSaWzSCZzWjYjgQ?usp=sharing>

LLM notes-

<https://drive.google.com/file/d/17D6dGyECqIsEx6jqCOJwnM7RKDvqirdq/view?usp=drive_link>

[Learn the fundamentals of cloud computing](https://github.com/learntocloud/learn-to-cloud)

[Software Development-For AI,](https://github.com/resources/articles/software-development)

[Introduction to Generative AI and Large Language Models](https://github.com/microsoft/generative-ai-for-beginners/blob/main/01-introduction-to-genai/README.md)

[aishwaryanr/awesome-generative-ai-guide](https://github.com/aishwaryanr/awesome-generative-ai-guide)

[Harness the power of generative AI for software ... - GitHub](https://github.com/readme/guides/coding-generative-ai)

[GoogleCloudPlatform/generative-ai](https://github.com/GoogleCloudPlatform/generative-ai)

[generative-model · GitHub Topics](https://github.com/topics/generative-model)

[**Hugging Face**](https://huggingface.co/)-AI Community

[lucidrains/vit-pytorch: Implementation of Vision Transformer ...](https://github.com/lucidrains/vit-pytorch)

[Transformers: State-of-the-art Machine Learning for Pytorch ...](https://github.com/huggingface/transformers)

[Transformers-LSTM-with-XGBRegressor](https://github.com/kaledhoshme123/Transformers-LSTM-with-XGBRegressor)

[tanikawa04/lstm-transformer](https://github.com/tanikawa04/lstm-transformer)

[Transformer with LSTM](https://www.kaggle.com/code/shujian/transformer-with-lstm)

[tf.keras.layers.LSTM | TensorFlow v2.16.1](https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM)

[Architecture and Working of Transformers in Deep Learning](https://www.geeksforgeeks.org/architecture-and-working-of-transformers-in-deep-learning/)

[Transformers in Machine Learning](https://www.geeksforgeeks.org/getting-started-with-transformers/)

[How Transformers Work: A Detailed Exploration of ...](https://www.datacamp.com/tutorial/how-transformers-work)

[Free AI tools from Google Cloud](https://cloud.google.com/use-cases/free-ai-tools?hl=en)

[12 Top Generative AI Tools (2024)](https://www.geeksforgeeks.org/popular-ai-tools-list/)

[Generative AI – What is it and How Does it Work?](https://www.nvidia.com/en-in/glossary/generative-ai/)

[Generative AI Concepts Course | DataCamp](https://www.datacamp.com/courses/generative-ai-concepts)

[GenAIOps with prompt flow and GitHub - Azure Machine ...](https://learn.microsoft.com/en-us/azure/machine-learning/prompt-flow/how-to-end-to-end-llmops-with-prompt-flow?view=azureml-api-2)

[llm-evaluation](https://github.com/topics/llm-evaluation)

[rasbt/LLMs-from-scratch: Implement a ChatGPT-like LLM in ...](https://github.com/rasbt/LLMs-from-scratch)

[large-language-models](https://github.com/topics/large-language-models)

[What is a Large Language Model (LLM)](https://www.geeksforgeeks.org/large-language-model-llm/)

[LLM? A Guide on Large Language Models and How They Work](https://www.googleadservices.com/pagead/aclk?sa=L&ai=DChcSEwjWqr_us_uJAxWgEIMDHQh4DrsYABAAGgJzZg&ae=2&aspm=1&co=1&ase=2&gclid=Cj0KCQiAgJa6BhCOARIsAMiL7V8sUSbv_tc3e4I6epU3CaLJ7LEs3dxKfqIS-lHbj4HXpBF2qoSYwkIaAr3hEALw_wcB&ohost=www.google.com&cid=CAESVuD2rb8XVYLIeSobcTVrbCPQAyOQvsqtoSPbETUhJvJuoI99DOELgs1t03DH3hPeo9emm5vgvzsGiVIVfgKSZYQtzLN6N8aJhMe8sGL_bJwIyZMc8c-k&sig=AOD64_3C6dlt2TBKy4Vn-jTSPZALHF1k8g&q&nis=4&adurl&ved=2ahUKEwjmsLvus_uJAxVmXWwGHbWDLucQ0Qx6BAgZEAE)

[**Large Language Models (LLMs) Concepts Course**](https://www.datacamp.com/courses/large-language-models-llms-concepts)

* [**How to Build LLM Applications with LangChain**](https://www.datacamp.com/tutorial/how-to-build-llm-applications-with-langchain)
* [**How to Train an LLM with PyTorch: A Step-By-Step Guide**](https://www.datacamp.com/tutorial/how-to-train-a-llm-with-pytorch)
* [**8 Top Open-Source LLMs for 2024 and Their Uses**](https://www.datacamp.com/blog/top-open-source-llms)
* [**Llama.cpp Tutorial: A Complete Guide to Efficient LLM Inference and Implementation**](https://www.datacamp.com/tutorial/llama-cpp-tutorial)
* [**Introduction to LangChain for Data Engineering & Data Applications**](https://www.datacamp.com/tutorial/introduction-to-langchain-for-data-engineering-and-data-applications)
* [**LlamaIndex: A Data Framework for the Large Language Models (LLMs) based applications**](https://www.datacamp.com/tutorial/llama-index-adding-personal-data-to-llms)
* [**How to Learn AI From Scratch in 2024: A Complete Expert Guide**](https://www.datacamp.com/blog/how-to-learn-ai)

[GPT (Generative Pre-trained Transformer)](https://www.geeksforgeeks.org/gpt-3-next-ai-revolution/)

[N-Gram Language Modelling with NLTK](https://www.geeksforgeeks.org/n-gram-language-modelling-with-nltk/)

[Introduction to N-grams in NLP](https://www.scaler.com/topics/nlp/ngrams-in-nlp/)

[asbt/LLMs-from-scratch: Implement a ChatGPT-like LLM in ...](https://github.com/rasbt/LLMs-from-scratch)

[How to Build a Large Language Model from Scratch Using ...](https://www.freecodecamp.org/news/how-to-build-a-large-language-model-from-scratch-using-python/)

[Zero-Shot vs One-Shot vs Few-Shot Learning](https://www.geeksforgeeks.org/zero-shot-vs-one-shot-vs-few-shot-learning/)

[Zero-Shot Prompting: Examples, Theory, Use Cases](https://www.datacamp.com/tutorial/zero-shot-prompting)

28-11-24

<https://drive.google.com/drive/folders/1fPET7nRlKAVGt7sI8MSaWzSCZzWjYjgQ?usp=drive_link>

[Hugging Face My Profile](https://huggingface.co/spaces)

[**Hugging Face**](https://huggingface.co/) [Spaces](https://huggingface.co/spaces)

[Retrieval Augmented Generation? A Guide to the Basics](https://www.datacamp.com/blog/what-is-retrieval-augmented-generation-rag)

[Advanced RAG on Hugging Face documentation using](https://huggingface.co/learn/cookbook/en/advanced_rag)

[LangChain](https://www.langchain.com/)

[LangChain docs](https://python.langchain.com/docs/introduction/)

[Conceptual guide | 🦜️🔗 LangChain](https://python.langchain.com/docs/concepts/)

<https://platform.openai.com/>

<https://huggingface.co/spaces/thisisdev/BillExtractor>

A screenshot of a computer

Description automatically generated

[Google AI Studio](https://aistudio.google.com/prompts/new_chat)

[Google AI | 🦜️🔗 LangChain](https://python.langchain.com/docs/integrations/llms/google_ai/)

AI Studio Key-https://aistudio.google.com/apikey

<https://python.langchain.com/docs/integrations/llms/google_ai/>

A screen shot of a computer

Description automatically generated

API key - AIzaSyA74XE\_86WfBYSWHSM-J-i9Xyx-9AN6hEk

[AlphaCode](https://alphacode.deepmind.com/) by Deepmind

03-12-24

<https://colab.research.google.com/drive/1iwF5EZp8orvyoUT9hOWz7ZAsGiQpwA01#scrollTo=y5Lbo_hqbiOk>

<https://drive.google.com/drive/folders/1fPET7nRlKAVGt7sI8MSaWzSCZzWjYjgQ?usp=drive_link>

[HuggingFace Documentation](https://huggingface.co/docs)



[What is Sentence Similarity?](https://huggingface.co/tasks/sentence-similarity)

[Models](https://huggingface.co/models?pipeline_tag=sentence-similarity)

[**Sentence Transformers**](https://huggingface.co/sentence-transformers)

[Serverless Inference API - Hugging Face](https://huggingface.co/docs/api-inference/en/index)

[Text-to-Speech AI: Lifelike Speech Synthesis](https://cloud.google.com/text-to-speech)

[What is Text-to-Speech?](https://huggingface.co/tasks/text-to-speech)

[**suno**](https://huggingface.co/suno)**/**[**bark-small**](https://huggingface.co/suno/bark-small)-https://huggingface.co/suno/bark-small

[facebook/mms-tts-eng](https://huggingface.co/facebook/mms-tts-eng)

[What is Automatic Speech Recognition?](https://huggingface.co/tasks/automatic-speech-recognition)

[Automatic speech recognition](https://huggingface.co/docs/transformers/en/tasks/asr)

[Download Sample Videos / Dummy Videos For Demo Use](https://www.sample-videos.com/)

[**Text-to-Speech AI**](https://cloud.google.com/text-to-speech#documentation)

04-12-24-Langchain

Drive link-

<https://drive.google.com/drive/folders/1fPET7nRlKAVGt7sI8MSaWzSCZzWjYjgQ?usp=drive_link>

Colab-<https://colab.research.google.com/drive/1Qm_FbtcEVFqIbZwunGjsyEMfJJupU5iI?usp=sharing>

Lonk-2-https://colab.research.google.com/drive/1Qm\_FbtcEVFqIbZwunGjsyEMfJJupU5iI?usp=sharing#scrollTo=rvSHVKoCbL5z

[Retrieval augmented generation (RAG)](https://python.langchain.com/docs/concepts/rag/)

[Chat models](https://python.langchain.com/docs/concepts/chat_models/)

[Streaming](https://python.langchain.com/docs/concepts/streaming/)

[ChatPromptTemplate](https://python.langchain.com/api_reference/core/prompts/langchain_core.prompts.chat.ChatPromptTemplate.html)

[ChatAnthropic](https://python.langchain.com/api_reference/anthropic/chat_models/langchain_anthropic.chat_models.ChatAnthropic.html)

[StrOutputParser](https://python.langchain.com/api_reference/core/output_parsers/langchain_core.output_parsers.string.StrOutputParser.html)

[Mistral AI Documentation](https://docs.mistral.ai/)

[Mistral AI APIs](https://console.mistral.ai/)

[State of AI Agents](https://www.langchain.com/stateofaiagents)

<https://huggingface.co/mistralai/Mistral-7B-v0.1>

[Langchain Documentaion](https://python.langchain.com/docs/introduction/)

[How to Build LLM Applications with LangChain Tutorial](https://www.datacamp.com/tutorial/how-to-build-llm-applications-with-langchain)

[Langchain](https://python.langchain.com/docs/integrations/vectorstores/pinecone/)

[Pinecone: The vector database to build knowledgeable AI](https://www.pinecone.io/)

[Pinecone Docs](https://docs.pinecone.io/integrations/langchain)

[Pinecone Documentation](https://docs.pinecone.io/guides/get-started/overview)

[Introduction to LangChain](https://www.geeksforgeeks.org/introduction-to-langchain/)

[langchain-community](https://pypi.org/project/langchain-community/)

[Chroma](https://www.trychroma.com/)

[Chroma Docs](https://docs.trychroma.com/)

[Cosine Similarity - GeeksforGeeks](https://www.geeksforgeeks.org/cosine-similarity/)

[The Top 7 Vector Databases in 2025](https://www.datacamp.com/blog/the-top-5-vector-databases)

[Top 10 LangChain Alternatives for Your Business - Rasa](https://rasa.com/blog/langchain-alternatives/" \l ":~:text=TensorFlow%20is%20a%20versatile%2C%20open,enterprises%20with%20substantial%20AI%20needs.)

[A screenshot of a computer

Description automatically generated](https://www.datacamp.com/blog/the-top-5-vector-databases)

Build a Retrieval Augmented Generation (RAG) App

<https://app.pinecone.io/>

Alternative to Langchain

1. [FlowiseAI](https://blog.apify.com/langchain-alternatives/#1-flowiseai)  
2. [Auto-GPT](https://blog.apify.com/langchain-alternatives/#2-auto-gpt)  
3. [AgentGPT](https://blog.apify.com/langchain-alternatives/#3-agentgpt)  
4. [BabyAGI](https://blog.apify.com/langchain-alternatives/#4-babyagi)  
5. [LangDock](https://blog.apify.com/langchain-alternatives/#5-langdock)  
6. [GradientJ](https://blog.apify.com/langchain-alternatives/#6-gradientj)  
7. [TensorFlow](https://blog.apify.com/langchain-alternatives/#7-tensorflow)  
8. [LlamaIndex](https://blog.apify.com/langchain-alternatives/#8-llamaindex)

Prompt Engineering

[Prompt Templates In LangChain](https://medium.com/@princekrampah/prompt-templates-in-langchain-248c015be3e0)

A computer screen with text and images

Description automatically generated

05-12-24-Promt Engineering

Prompt Engineering

Notes-<https://docs.google.com/document/d/1e5UdugYiEHcK7uKK8oKgZzb_y4TsY9pI52AaQDD0Z0A/edit?usp=sharing>

<https://docs.google.com/document/d/1e5UdugYiEHcK7uKK8oKgZzb_y4TsY9pI52AaQDD0Z0A/edit?usp=sharing>

[Prompt Engineering for AI Guide](https://cloud.google.com/discover/what-is-prompt-engineering)

[Prompt engineering - OpenAI API](https://platform.openai.com/docs/guides/prompt-engineering)

[What is Prompt Engineering? A Detailed Guide For 2024](https://www.datacamp.com/blog/what-is-prompt-engineering-the-future-of-ai-communication)

[Awesome-Prompt-Engineering](https://github.com/promptslab/Awesome-Prompt-Engineering)

<https://github.com/langflow-ai/langflow>

[Prompt Engineering Guide](https://www.promptingguide.ai/)

[Prompting Techniques](https://www.promptingguide.ai/techniques)

[How to Become a Prompt Engineer: A Comprehensive Guide](https://www.datacamp.com/blog/how-to-become-a-prompt-engineer)

[Understanding oLLaMa](https://www.hostinger.in/tutorials/what-is-ollama#:~:text=and%20enhance%20performance.-,Local%20AI%20model%20management,who%20prioritize%20strict%20data%20security.)

[LoRA](https://huggingface.co/docs/diffusers/en/training/lora)

[rag-fusion](https://python.langchain.com/v0.1/docs/templates/rag-fusion/)

**[LM StudioDocs](https://lmstudio.ai/)**

[Prompt Engineering with LangChain](https://www.datacamp.com/tutorial/prompt-engineering-with-langchain)

[ConversationBufferMemory](https://python.langchain.com/api_reference/langchain/memory/langchain.memory.buffer.ConversationBufferMemory.html)

[LangChain Python API Reference](https://python.langchain.com/api_reference/index.html)

[Introduction to Using Transformers and Hugging Face](https://www.datacamp.com/tutorial/an-introduction-to-using-transformers-and-hugging-face)

A black rectangle with white text

Description automatically generated

**Tools for Prompt Design and Optimization**

1. **OpenAI API**: [OpenAI API](https://platform.openai.com/)
2. **Hugging Face Transformers**: Hugging Face Transformers
3. **LangChain**: [LangChain](https://www.langchain.com/)
4. **PromptPerfect**: PromptPerfect

**Tools for Prompt Testing and Debugging**

1. **WeightWatcher**: [WeightWatcher GitHub Repository](https://github.com/CalculatedContent/WeightWatcher)
2. **EleutherAI's GPT-Neo**: [EleutherAI](https://www.eleuther.ai/)

**Tools for Visualization and Workflow Integration**

1. **Streamlit**: [Streamlit](https://streamlit.io/)
2. **Gradio**: [Gradio](https://gradio.app/)

**Technologies for Advanced Prompt Engineering**

1. **Google Gemini**: [Google Gemini](https://inthecloud.withgoogle.com/)​

[KDnuggets](https://www.kdnuggets.com/3-new-prompt-engineering-resources)

.

1. **LLMLingua**: [LLMLingua GitHub](https://github.com/)​

[KDnuggets](https://www.kdnuggets.com/3-new-prompt-engineering-resources)

.

1. **ChatGPT Plugins**: [ChatGPT Plugins](https://openai.com/chatgpt/plugins/)

**Tools for Fine-Tuning and Customization**

1. **PyTorch**: [PyTorch](https://pytorch.org/)
2. **TensorFlow**: [TensorFlow](https://www.tensorflow.org/)
3. **AI21 Studio**: AI21 Studio

**Specialized Prompt Libraries and Repositories**

1. **PromptBase**: [PromptBase](https://promptbase.com/)
2. **OpenPrompt**: [OpenPrompt GitHub](https://github.com/thunlp/OpenPrompt)

**Types and Techniques of Prompt Engineering**

Prompt Engineering involves designing tailored inputs for AI models to get accurate and effective outputs. Understanding its **types** and **techniques** will help you optimize the performance of AI systems.

**Types of Prompts in Prompt Engineering**

**1. Instruction-Based Prompts**

These prompts give direct commands or instructions to the AI.

* **Examples:**
  + "Summarize this article in 3 sentences."
  + "Explain Newton's laws in simple terms."

**2. Role-Based Prompts**

Assign a role to the AI to narrow its behavior and style.

* **Examples:**
  + "You are a math tutor. Explain the Pythagorean theorem."
  + "Act as a project manager and create a risk management plan."

**3. Contextual Prompts**

Provide additional context or background information to the model to generate better responses.

* **Examples:**
  + "Given the rise in electric vehicles, explain how lithium demand will impact the economy."
  + "Using this data [insert table], write an analysis of sales trends."

**4. Example-Driven Prompts**

Also called **few-shot prompts**, these include examples to set a pattern for the AI to follow.

* **Examples:**
  + "Translate the following into French:
    - Hello -> Bonjour
    - Good morning -> Bon matin
    - Goodbye -> [Your response]"

**5. Chain-of-Thought Prompts**

Encourage the AI to think step-by-step for solving complex problems.

* **Examples:**
  + "Solve this problem step-by-step: If a train travels 50 miles in 1 hour, how long will it take to travel 200 miles?"

**6. Open-Ended Prompts**

Used for generating creative or exploratory outputs without strict constraints.

* **Examples:**
  + "Write a short story about a dragon who learns to fly."
  + "Generate ideas for a sustainable startup."

**7. Specific Prompts**

Narrowly focused prompts to control the output format, style, or content.

* **Examples:**
  + "List the advantages and disadvantages of cloud computing in a table format."
  + "Write a professional email apologizing for a delay in delivery."

**8. Iterative Prompts**

Involves using follow-up questions or refining previous outputs to reach the desired result.

* **Example:**
  + First Prompt: "Summarize this article."
  + Second Prompt: "Rewrite the summary to make it more engaging."

**9. Multimodal Prompts**

Used for AI systems that can handle both text and other input types (e.g., images, audio).

* **Example:**
  + "Describe the key features of this image [upload image]."

**Techniques in Prompt Engineering**

**1. Few-Shot Learning**

Provide a few examples in the prompt to train the AI on the desired output style and structure.

* **Example:**
  + "Classify the following as 'Positive' or 'Negative':
    - The weather is amazing. -> Positive
    - I feel terrible today. -> Negative
    - The movie was fantastic. -> [Your response]"

**2. Zero-Shot Learning**

Ask the AI to perform a task without providing any examples, relying entirely on the prompt's clarity.

* **Example:**
  + "Write a haiku about the moon."

**3. Chain-of-Thought (CoT) Prompting**

Encourage the model to reason step-by-step for complex problems.

* **Example:**
  + "If a worker builds 5 chairs in a day, how many chairs will 3 workers build in 4 days? Break your answer into steps."

**4. Iterative Prompt Refinement**

Improve results by analyzing and tweaking the prompt multiple times.

* **Steps:**
  1. Test the prompt.
  2. Identify shortcomings in the output.
  3. Add context, constraints, or examples.

**5. Role Specification**

Assign a persona to the AI for targeted outputs.

* **Example:**
  + "You are a historian. Write about the significance of the French Revolution."

**6. Multimodal Prompting**

Incorporate non-text data (images, graphs) for AI systems like GPT-4V or DALL·E.

* **Example:**
  + "Analyze this chart and summarize its key trends: [chart link]."

**7. Contextualization**

Provide a detailed background to improve relevance.

* **Example:**
  + "Given this scenario: A company faces declining sales due to competition. Suggest 3 strategies to increase market share."

**8. Task Decomposition**

Break complex problems into smaller tasks and tackle them sequentially.

* **Example:**
  1. "List the main points in the article."
  2. "Expand each point into a paragraph."
  3. "Write a conclusion based on the paragraphs."

**9. Instruction Tuning**

Use explicit instructions for fine-grained control over output.

* **Example:**
  + "Write a Python function to reverse a string. Include comments explaining each step."

**10. Formatting and Constraints**

Guide the AI by specifying structure or length.

* **Examples:**
  + "Generate a 100-word summary of this text."
  + "List 5 bullet points about the advantages of solar energy."

**11. Ethical and Bias-Aware Prompting**

Design prompts that reduce bias or lead to neutral, inclusive outputs.

* **Example:**
  + Avoid: "Why are women better at multitasking?"
  + Better: "Explain the science behind multitasking and how it varies among individuals."

**12. Temperature Control in API Prompting**

Adjust the **temperature parameter** in API-based models to control randomness:

* Lower values (e.g., 0.2): More deterministic and focused responses.
* Higher values (e.g., 0.8): Creative but less predictable outputs.