

LLM Course

[LLM Course documentation](#)

[Introduction](#)

[LLM Course](#)

■ [View all resources](#)

[Agents Course](#)

[Audio Course](#)

[Community Computer Vision Course](#)

[Deep RL Course](#)

[Diffusion Course](#)

[LLM Course](#)

[MCP Course](#)

[ML for 3D Course](#)

[ML for Games Course](#)

[Open-Source AI Cookbook](#)

[Robotics Course](#)

[a smol course](#)

[Search documentation](#)

[AR](#)

[BN](#)

[DE](#)

[EN](#)

[ES](#)

FA

FR

GJ

HE

HI

ID

IT

JA

KO

MY

NE

PL

PT

RO

RU

RUM

TE

TH

TR

VI

ZH-CN

ZH-TW

Join the Hugging Face community

and get access to the augmented documentation experience

Collaborate on models, datasets and Spaces

Faster examples with accelerated inference

Switch between documentation themes

Sign Up

to get started

Copy page

Introduction

Welcome to the **LLM** Course!

This course will teach you about large language models (LLMs) and natural language processing (NLP) using libraries from the

Hugging Face

ecosystem —

LLM Transformers

,

LLM Datasets

,

LLM Tokenizers

, and

LLM Accelerate

— as well as the

Hugging Face Hub

.

We'll also cover libraries outside the Hugging Face ecosystem. These are amazing contributions to the AI community and incredibly useful tools.

It's completely free and without ads.

Understanding NLP and LLMs

While this course was originally focused on NLP (Natural Language Processing), it has evolved to emphasize Large Language Models (LLMs), which represent the latest advancement in the field.

What's the difference?

NLP (Natural Language Processing)

is the broader field focused on enabling computers to understand, interpret, and generate human language. NLP encompasses many techniques and tasks such as sentiment analysis, named entity recognition, and machine translation.

LLMs (Large Language Models)

are a powerful subset of NLP models characterized by their massive size, extensive training data, and ability to perform a wide range of language tasks with minimal task-specific training. Models like the Llama, GPT, or Claude series are examples of LLMs that have revolutionized what's possible in NLP.

Throughout this course, you'll learn about both traditional NLP concepts and cutting-edge LLM techniques, as understanding the foundations of NLP is crucial for working effectively with LLMs.

What to expect?

Here is a brief overview of the course:

Chapters 1 to 4 provide an introduction to the main concepts of the **■** Transformers library. By the end of this part of the course, you will be familiar with how Transformer models work and will know how to use a model from the

Hugging Face Hub

, fine-tune it on a dataset, and share your results on the Hub!

Chapters 5 to 8 teach the basics of **■** Datasets and **■** Tokenizers before diving into classic NLP tasks and LLM techniques. By the end of this part, you will be able to tackle the most common language processing challenges by yourself.

Chapter 9 goes beyond NLP to cover how to build and share demos of your models on the **■** Hub. By the end of this part, you will be ready to showcase your **■** Transformers application to the world!

Chapters 10 to 12 dive into advanced LLM topics like fine-tuning, curating high-quality datasets, and building reasoning models.

This course:

Requires a good knowledge of Python

Is better taken after an introductory deep learning course, such as

fast.ai's

Practical Deep Learning for Coders

or one of the programs developed by

DeepLearning.AI

Does not expect prior

PyTorch

or

TensorFlow

knowledge, though some familiarity with either of those will help

After you've completed this course, we recommend checking out DeepLearning.AI's

Natural Language Processing Specialization

, which covers a wide range of traditional NLP models like naive Bayes and LSTMs that are well worth knowing about!

Who are we?

About the authors:

Abubakar Abid

completed his PhD at Stanford in applied machine learning. During his PhD, he founded

Gradio

, an open-source Python library that has been used to build over 600,000 machine learning demos. Gradio was acquired by Hugging Face, which is where Abubakar now serves as a machine learning team lead.


Ben Burtenshaw

is a Machine Learning Engineer at Hugging Face. He completed his PhD in Natural Language Processing at the University of Antwerp, where he applied Transformer models to generate children stories for the purpose of improving literacy skills. Since then, he has focused on educational materials and tools for the wider community.


Matthew Carrigan

is a Machine Learning Engineer at Hugging Face. He lives in Dublin, Ireland and previously worked as an ML engineer at Parse.ly and before that as a post-doctoral researcher at Trinity College Dublin. He does not believe we're going to get to AGI by scaling existing architectures, but has high hopes for robot immortality regardless.

Lysandre Debut

is a Machine Learning Engineer at Hugging Face and has been working on the  Transformers library since the very early development stages. His aim is to make NLP accessible for everyone by developing tools with a very simple API.

Sylvain Gugger

is a Research Engineer at Hugging Face and one of the core maintainers of the  Transformers library. Previously he was a Research Scientist at fast.ai, and he co-wrote

Deep Learning for Coders with fastai and PyTorch

with Jeremy Howard. The main focus of his research is on making deep learning more accessible, by designing and improving techniques that allow models to train fast on limited resources.

Dawood Khan

is a Machine Learning Engineer at Hugging Face. He's from NYC and graduated from New York University studying Computer Science. After working as an iOS Engineer for a few years, Dawood quit to start Gradio with his fellow co-founders. Gradio was eventually acquired by Hugging Face.

Merve Noyan

is a developer advocate at Hugging Face, working on developing tools and building content around them to democratize machine learning for everyone.

Lucile Saulnier

is a machine learning engineer at Hugging Face, developing and supporting the use of open source tools. She is also actively involved in many research projects in the field of Natural Language Processing such as collaborative training and BigScience.

Lewis Tunstall

is a machine learning engineer at Hugging Face, focused on developing open-source tools and making them accessible to the wider community. He is also a co-author of the O'Reilly book

Natural Language Processing with Transformers

.

Leandro von Werra

is a machine learning engineer in the open-source team at Hugging Face and also a co-author of the O'Reilly book

Natural Language Processing with Transformers

. He has several years of industry experience bringing NLP projects to production by working across the whole machine learning stack..

FAQ

Here are some answers to frequently asked questions:

Does taking this course lead to a certification?

Currently we do not have any certification for this course. However, we are working on a certification program for the Hugging Face ecosystem — stay tuned!

How much time should I spend on this course?

Each chapter in this course is designed to be completed in 1 week, with approximately 6-8 hours of work per week. However, you can take as much time as you need to complete the course.

Where can I ask a question if I have one?

If you have a question about any section of the course, just click on the "

Ask a question

" banner at the top of the page to be automatically redirected to the right section of the

Hugging Face forums

:

Note that a list of

project ideas

is also available on the forums if you wish to practice more once you have completed the course.

Where can I get the code for the course?

For each section, click on the banner at the top of the page to run the code in either Google Colab or Amazon SageMaker Studio Lab:

The Jupyter notebooks containing all the code from the course are hosted on the

`huggingface/notebooks`

repo. If you wish to generate them locally, check out the instructions in the

course

repo on GitHub.

How can I contribute to the course?

There are many ways to contribute to the course! If you find a typo or a bug, please open an issue on the

course

repo. If you would like to help translate the course into your native language, check out the instructions

here

.

What were the choices made for each translation?

Each translation has a glossary and

`TRANSLATING.txt`

file that details the choices that were made for machine learning jargon etc. You can find an example for German

here

.

Can I reuse this course?

Of course! The course is released under the permissive

Apache 2 license

. This means that you must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. If you would like to cite the course, please use the following BibTeX:

Copied

@misc

{huggingfacecourse,

author = {Hugging Face}

,

title =

{The Hugging Face Course, 2022}

,

howpublished = "\url

{<https://huggingface.co/course>}

",

year =

{2022}

,

note = "[Online; accessed

<

today

>

]"

}

Languages and translations

Thanks to our wonderful community, the course is available in many languages beyond English ■!
Check out the table below to see which languages are available and who contributed to the translations:

Language

Authors

French

@lbourdois

,

@ChainYo

,

@melaniedrevet

,

@abdouaziz

Vietnamese

@honghanhh

Chinese (simplified)

@zh1hyx

,

petrichor1122

,

@yaoqih

Bengali

(WIP)

@avishek-018

,

@eNipu

German

(WIP)

@JesperDramsch

,

@MarcusFra

,

@fabridamicelli

Spanish

(WIP)

@camartinezbu

,

@munozariasjm

,

@fordaz

Persian

(WIP)

@jowharshamshiri

,

@schoobani

Gujarati

(WIP)

@pandyaved98

Hebrew

(WIP)

@omer-dor

Hindi

(WIP)

@pandyaved98

Bahasa Indonesia

(WIP)

@gstdl

Italian

(WIP)

@CaterinaBi

,

@ClonedOne

,

@Nolanogenn

,

@EdAbati

,

@gdacciaro

Japanese

(WIP)

@hiromu166

,

@younesbelkada

,

@HiromuHota

Korean

(WIP)

@Doohae

,

@wonhyeongseo

,

@dlfrnaos19

Portuguese

(WIP)

@johnnv1

,

@victorescosta

,

@LincolnVS

Russian

(WIP)

@pdumin

,

@svv73

Thai

(WIP)

@peeraponw

,

@a-krirk

,

@jomariya23156

,

@ckingkan

Turkish

(WIP)

@tanersekmen

,

@mertbozkir

,

@ftarlaci

,

@akkasayaz

Chinese (traditional)

(WIP)

@davidpeng86

For some languages, the

course YouTube videos

have subtitles in the language. You can enable them by first clicking the

CC

button in the bottom right corner of the video. Then, under the settings icon ■■, you can select the language you want by selecting the

Subtitles/CC

option.

Don't see your language in the above table or you'd like to contribute to an existing translation? You can help us translate the course by following the instructions

here

.

Let's go ■

Are you ready to roll? In this chapter, you will learn:

How to use the

`pipeline()`

function to solve NLP tasks such as text generation and classification

About the Transformer architecture

How to distinguish between encoder, decoder, and encoder-decoder architectures and use cases

Update

on GitHub

←

Introduction

Natural Language Processing and Large Language Models

→