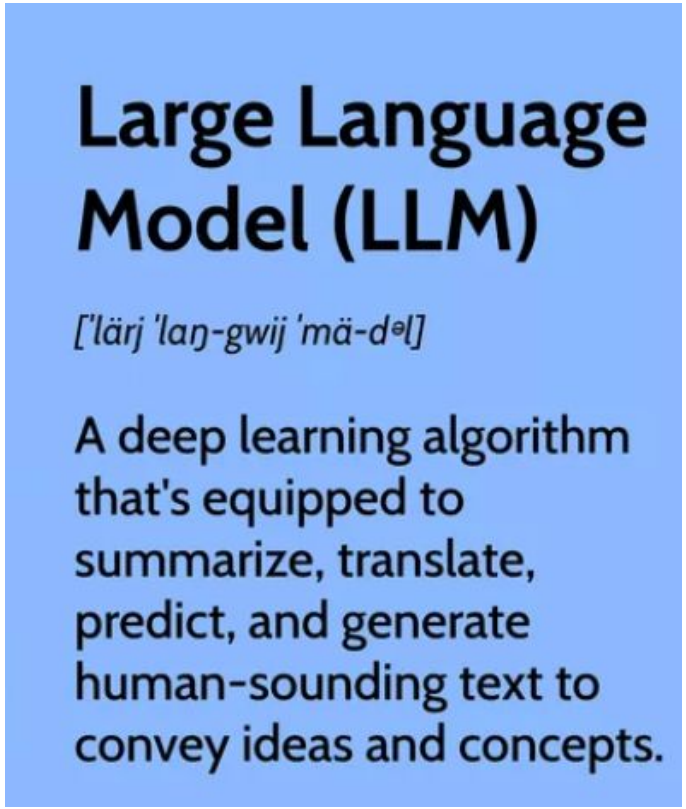
Unstructured to Structured using LLM

Priya Basker & Theodoros Manassis & Jacob HP

Table of contents

LLM



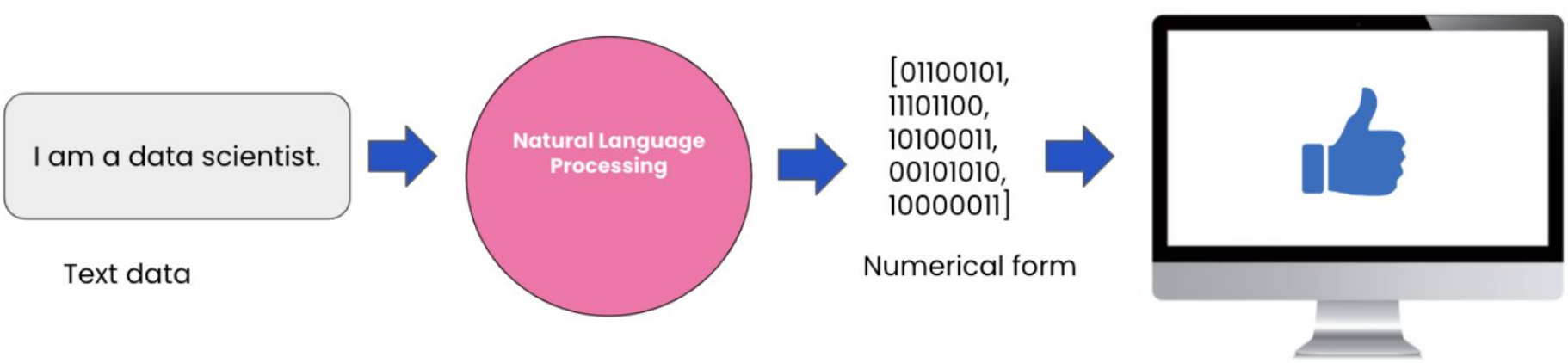
KEY TAKEAWAYS

- Large language models utilize deep learning algorithms to recognize, interpret, and generate human-sounding language.
- A large language model utilizes massive datasets, often featuring 100 million or more parameters, in order to solve common language problems.
- Developed by OpenAI, **ChatGPT is one of the most recognizable large language models. Google's BERT, Facebook's LLaMA, and Anthropic's Claude 2 are other examples of LLMs.**
- Example of open source LLMs are **PaLM, LaMDA, GPT, BERT, XLM** etc.
- Examples of closed source LLMs are **HyperCLOVA, GoPHER, Chinchilla** etc.
- Some of the ways in which large language models are used include content creation, translation, code generation for developers, audio transcription, and virtual chat or assistant applications.

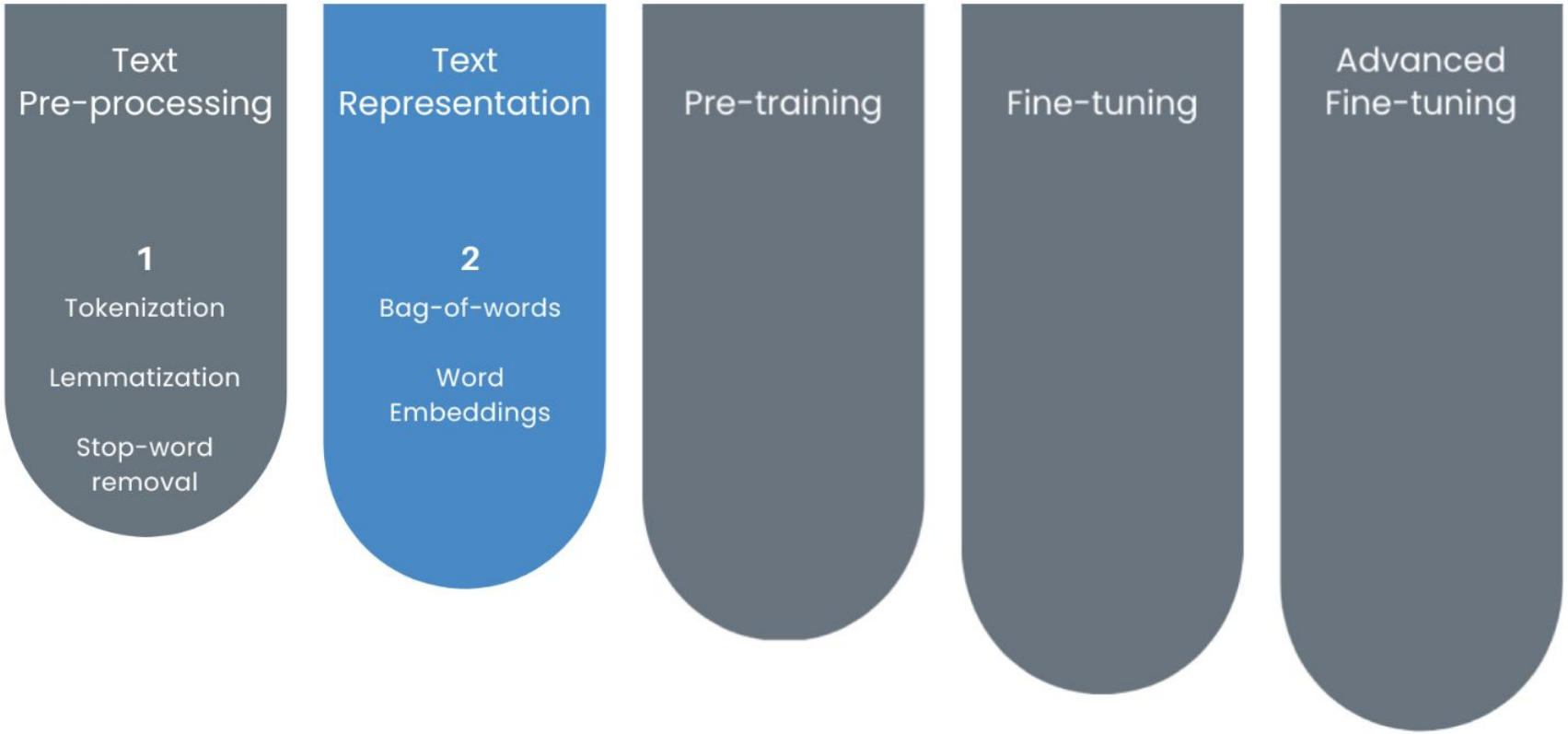
How do LLMs understand

- Trained on vast amounts of data
- Largeness of LLMs: parameters
- Parameters represent the patterns and rules
- More parameters -> complex patterns
- Generates sophisticated and accurate responses

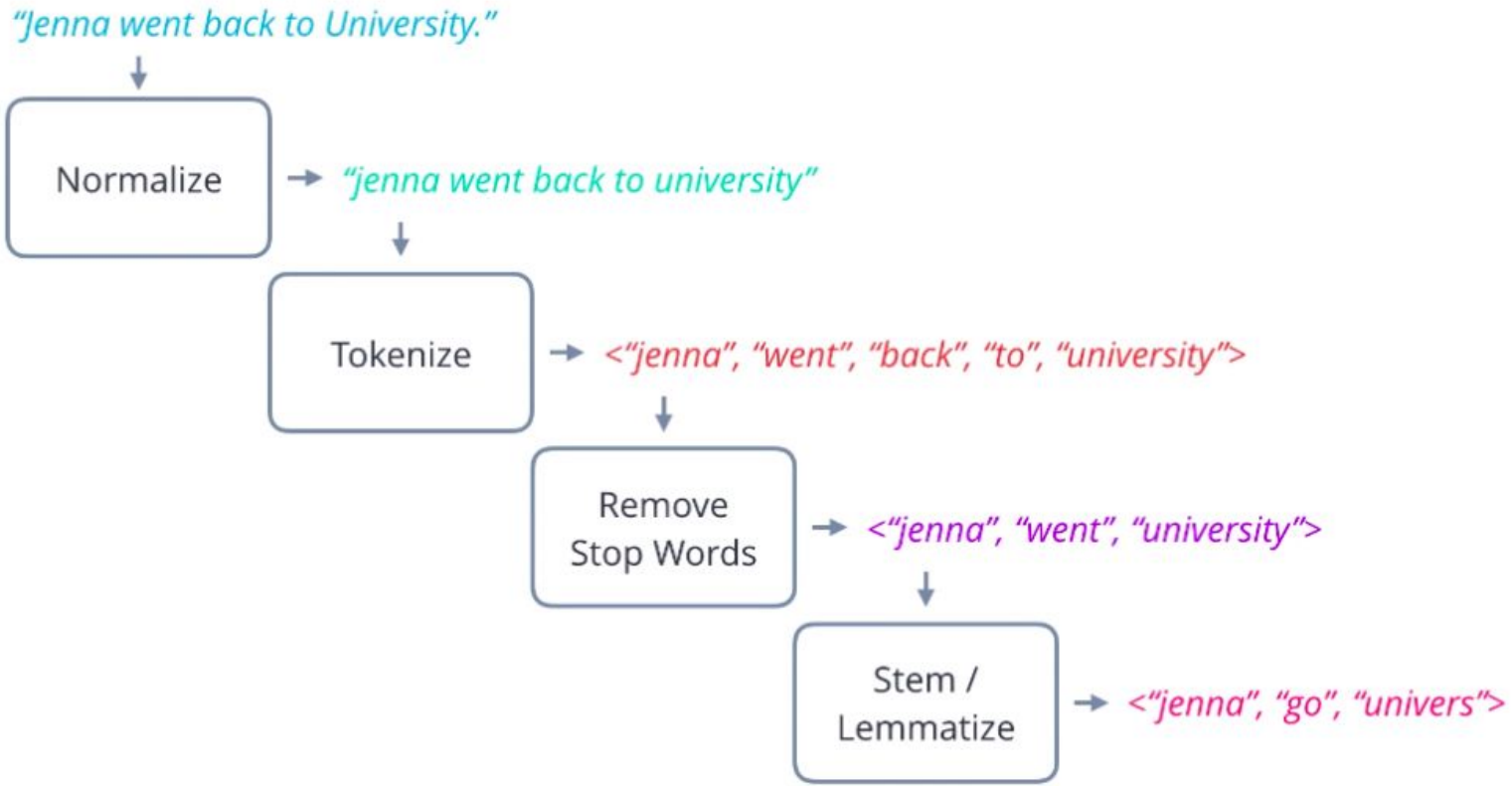
NLP


I am a data scientist.
Text data
Natural Language Processing
[01100101,
11101100,
10100011,
00101010,
10000011]
Numerical form

Building blocks of LLM



Text Preprocessing

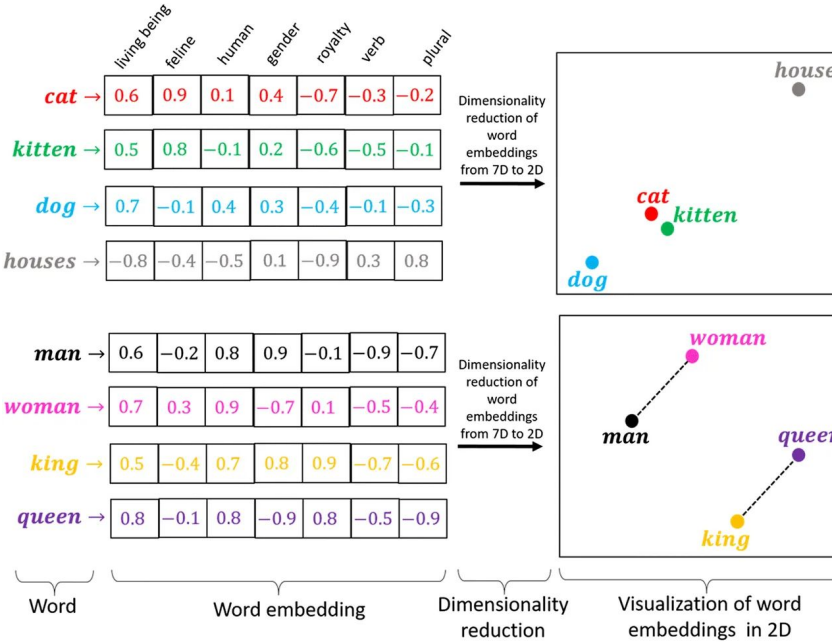


Text Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

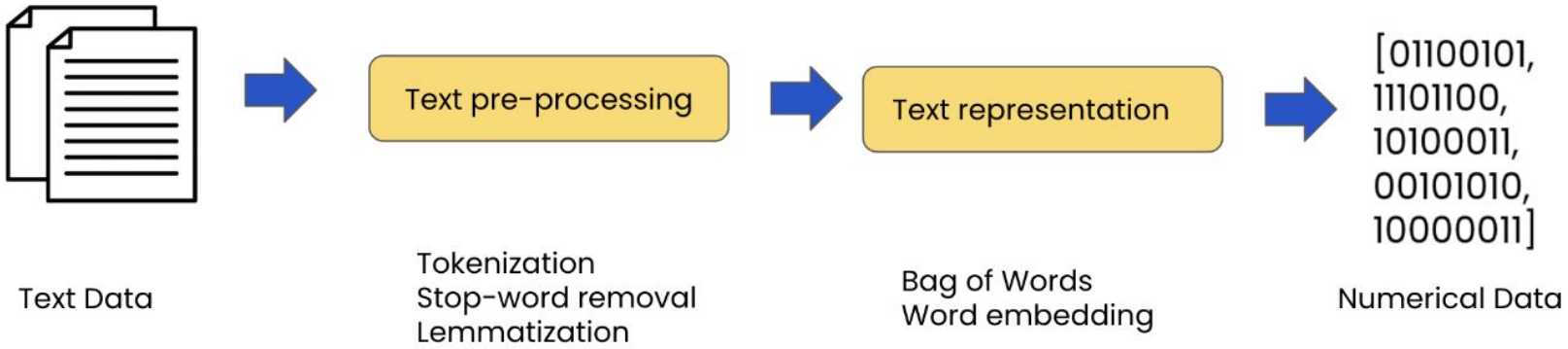
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

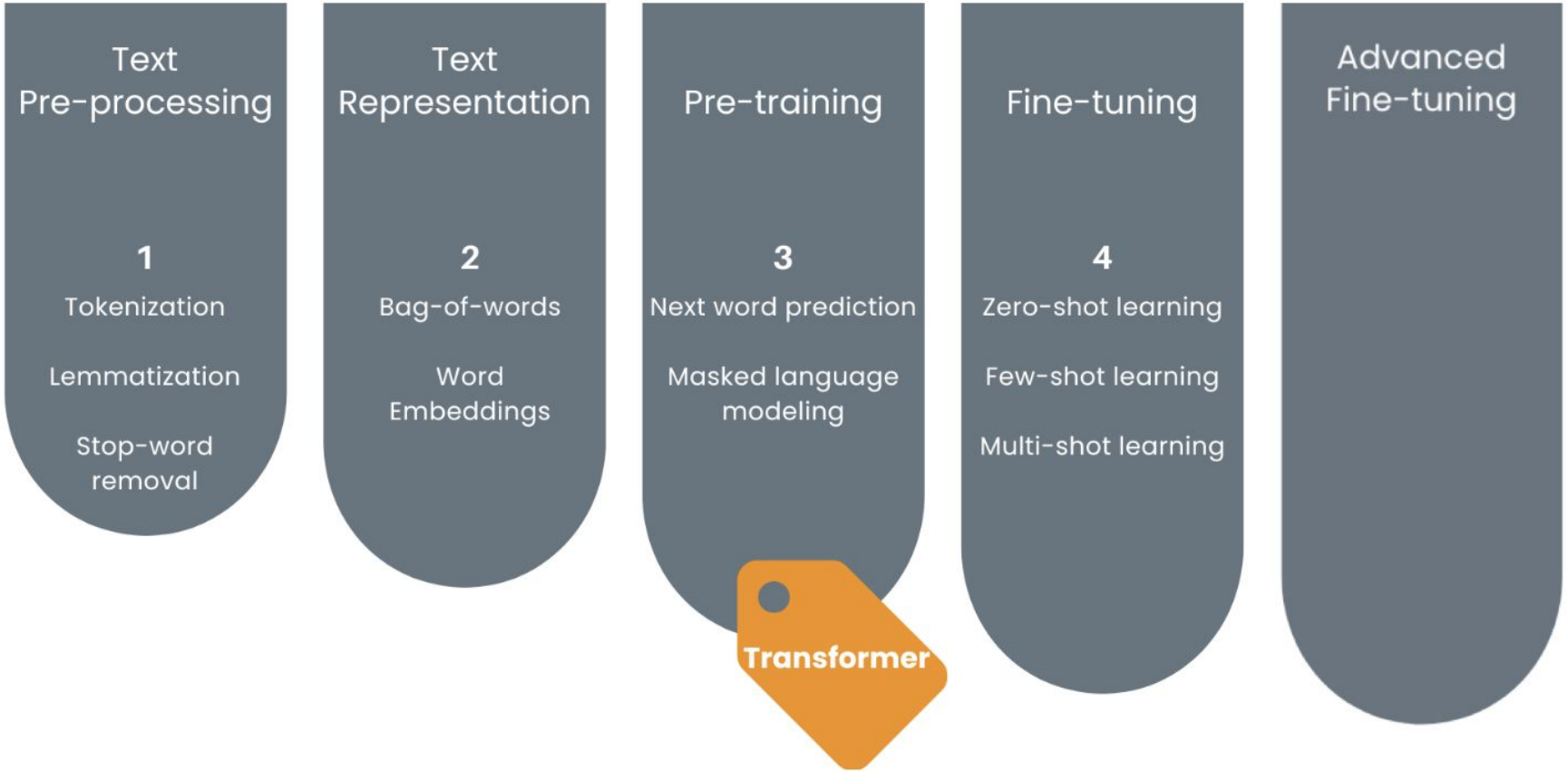
Bag of words
https://koushik1102.medium.com/nlp-bag-of-words-and-tf-idf-explained-fd1f49dce7c4



Word-to-Vec
https://medium.com/@hari4om/word-embedding-d816f643140

- Convert pre-processed text to numerical format




Text Pre-processing
1
Tokenization
Lemmatization
Stop-word removal
Text Representation
2
Bag-of-words
Word Embeddings
Pre-training
3
Next word prediction
Masked language modeling
Transformer
Fine-tuning
4
Zero-shot learning
Few-shot learning
Multi-shot learning
Advanced Fine-tuning

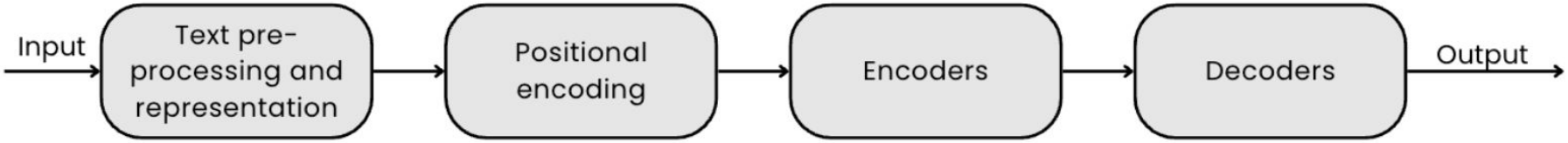
Generative pre-training

- Trained using generative pre-training
  - Input data of text tokens
  - Trained to predict the tokens within the dataset

- Types:
  - Next word prediction
  - Masked language modeling

Inside the transformer

- **Input:** Jane, who lives in New York and works as a software



Preprocessing :
Tokenization,
Stopword removal
Lemmatization

Text representation:
Word embedding

Information the position of the word

Understanding the distant word

Attention mechanism directs attention to specific words and relationship

Neural network - different layers interprets different features to learn complex and passes to next layer

Uses Attention mechanism and neural networks to provide outputs.

Transformers and long-range dependencies

- **Initial challenge**: long-range dependency
- **Attention**: focus on different parts of the input

- **Example**: "Jane, who lives in New York and works as a software engineer, loves exploring new restaurants in the city."
- "Jane" --- "loves exploring new restaurants"

Self-attention and multi-head attention

Self-attention

- Weighs the importance of each word
- Captures long-range dependencies

Multi-head attention

- Next level of self-attention
- Splits input into multiple heads with each head focusing on different aspects

Attention in a party

- **Attention:** Self and multi-head

- **Example:**
  - Group conversation at a party
  - Selective attention to relevant speaker
  - Filter noise

Self-attention

- Focus on each person's words
- Evaluate and compare their relevance
- Weigh each speaker's input
- Combines for a comprehensive understanding

Multi-head attention

- Split attention into "**multiple**" channels
- Focus on different aspects of conversation
- Speaker's emotions, primary topic, and related side-topics
- Process each aspect and merge

Processes multiple parts simultaneously

- **Limitation of traditional language models:**
  - Sequential - one word at a time

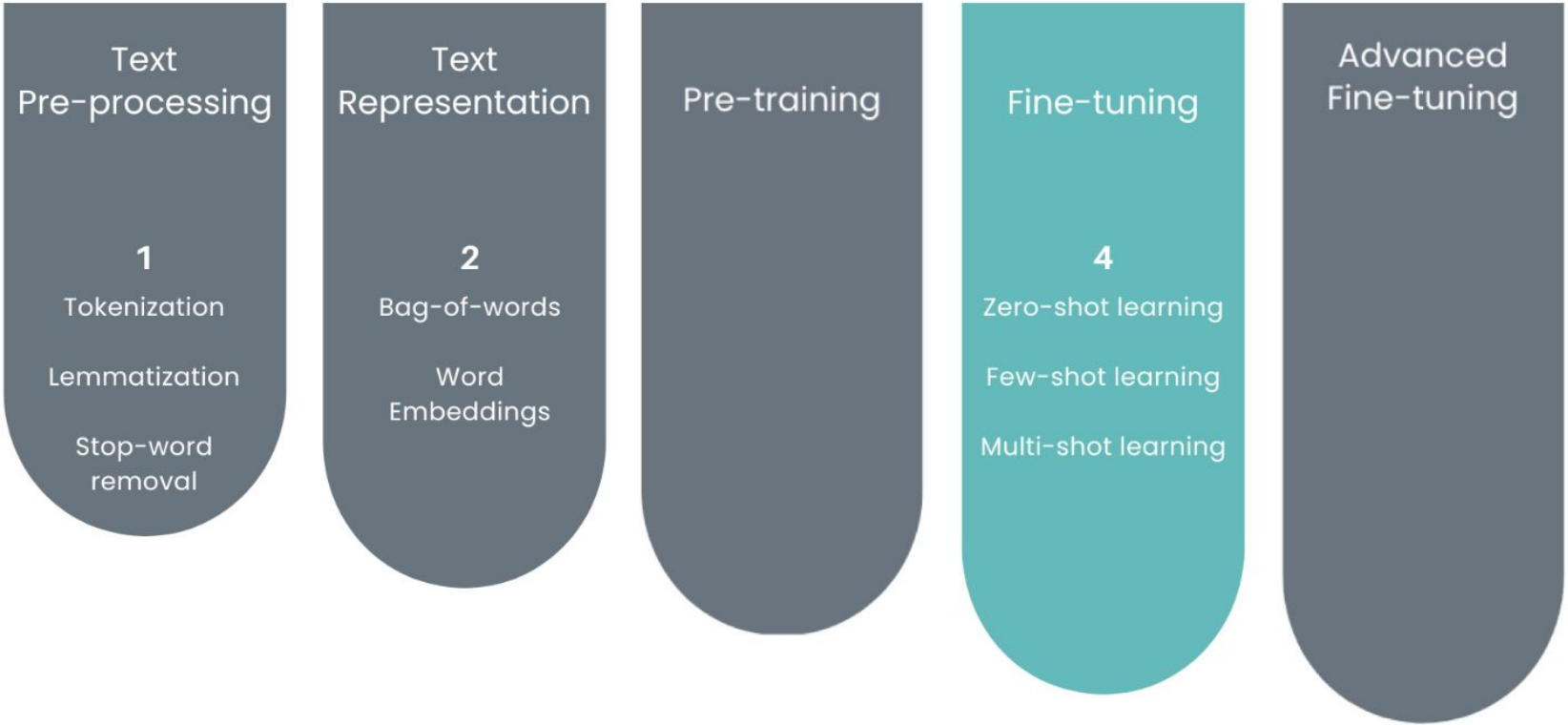
- **Transformers:**
  - Process multiple parts simultaneously
  - Faster processing

- **For example:**
  - "The cat sat on the mat"
  - Processes "cat," "sat," "on," "the," and "mat" at the same time

Multi-head attention advantages

- "The boy went to the store to buy some groceries, and he found a discount on his favorite cereal."

- **Attention**: "boy," "store," "groceries," and "discount"
- **Self-attention**: "boy" and "he" -> same person
- **Multi-head attention**: multiple channels
  - Character ("boy")
  - Action ("went to the store," "found a discount")
  - Things involved ("groceries," "cereal")


Text Pre-processing
1
Tokenization
Lemmatization
Stop-word removal
Text Representation
2
Bag-of-words
Word Embeddings
Pre-training
Fine-tuning
4
Zero-shot learning
Few-shot learning
Multi-shot learning
Advanced Fine-tuning

Fine Tuning

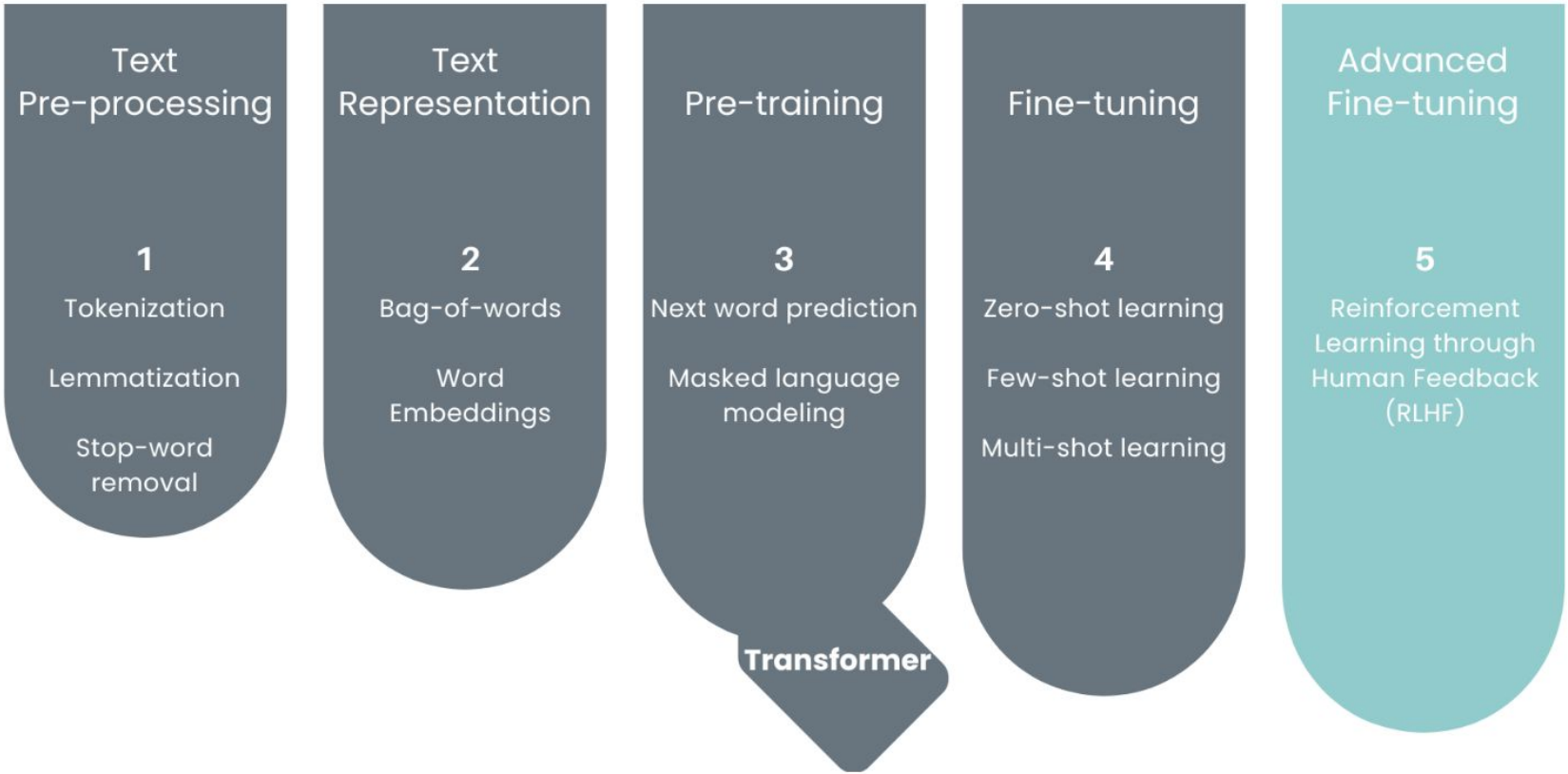
Zero-Shot Learning	The Model cannot look at any examples from the target class during training
One-Shot Learning	The Model observes one example from the target class during training
k-Shot Learning	The Model observes k-examples from the target class during training

https://prachi-gopalani.medium.com/zero-shot-few-shot-one-shot-learning-in-nlp-341aa684cdb2

Fine-tuning vs. Pre-training

- **Fine-tuning**
- Compute
  - 1-2 CPU and GPU
- Training time
  - Hours to days
- Data
  - ~1 gigabyte

- **Pre-training**
- Compute
  - Thousands of CPUs and GPUs
- Training time
  - Weeks to months
- Data
  - Hundreds of gigabytes


Text Pre-processing
1
Tokenization
Lemmatization
Stop-word removal
Text Representation
2
Bag-of-words
Word Embeddings
Pre-training
3
Next word prediction
Masked language modeling
Transformer
Fine-tuning
4
Zero-shot learning
Few-shot learning
Multi-shot learning
Advanced Fine-tuning
5
Reinforcement Learning through Human Feedback (RLHF)

But, why RLHF?

- General-purpose training data lacks quality
  - Noise
  - Errors
  - Inconsistencies
  - Reduced accuracy

Example of reduced accuracy:

- Trained on data from online discussion forums
- Unvalidated opinions and facts
- Needs external expert validation

- Model output reviewed by human
- Updates model based on the feedback

- Step 1:
  - Receives a prompt
  - Generates multiple responses

- Step 2:
  - Human expert checks these responses
  - Ranks the responses based on quality
    - Accuracy
    - Relevance
    - Coherence

- Step 3:
  - Learns from expert's ranking
  - To align its response in future with their preferences

Simple Comparison between previous models and LLM

Challenges of language modeling:

- Word sequences
- Understanding context
- Long-range dependency

Single-task learning:

- Task-specific
- Less flexible
- Traditional models and early LLMs

Multi-task learning:

- Versatile
- Multiple tasks
- More developed LLMs

SageMaker

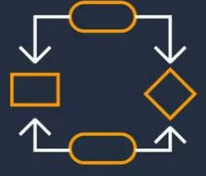
Amazon SageMaker

Addressing challenges to machine learning

First fully integrated development environment (IDE) for machine learning
Amazon SageMaker Studio

Enhanced notebook experience with quick-start & easy collaboration
Amazon SageMaker Notebooks (Preview)

Experiment management system to organize, track & compare thousands of experiments
Amazon SageMaker Experiments



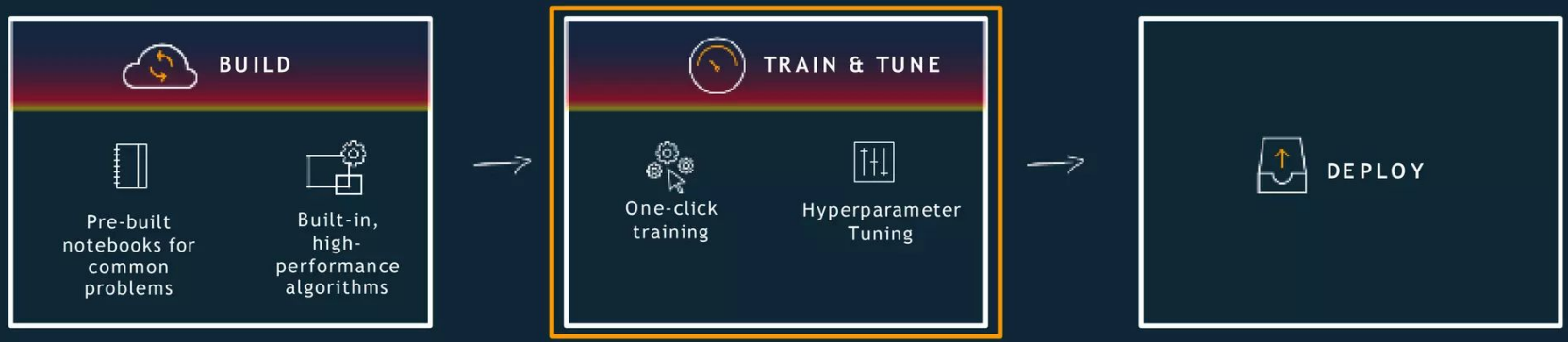
Automatic debugging, analysis, and alerting
Amazon SageMaker Debugger

Model monitoring to detect deviation in quality & take corrective actions
Amazon SageMaker Model Monitor

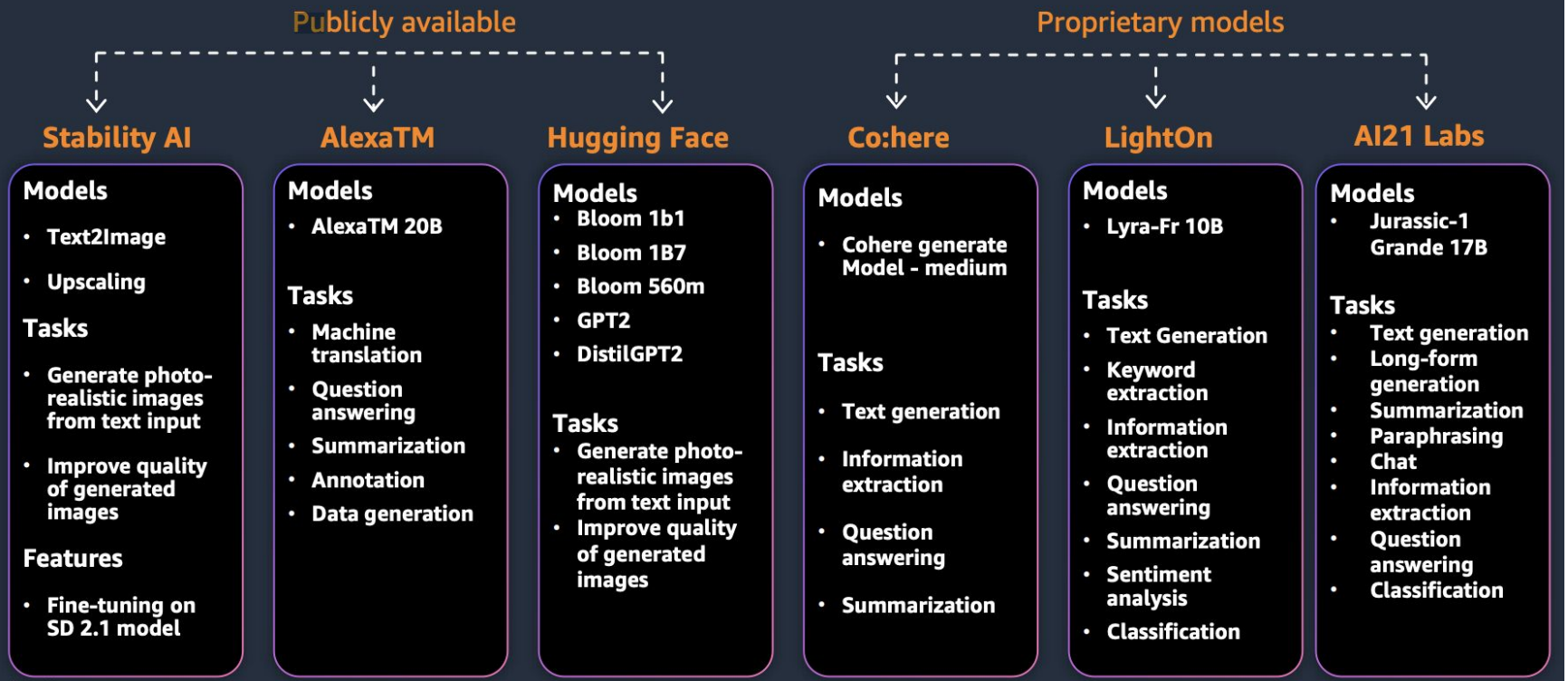
Automatic generation of ML models with full visibility & control
Amazon SageMaker Autopilot

Amazon SageMaker

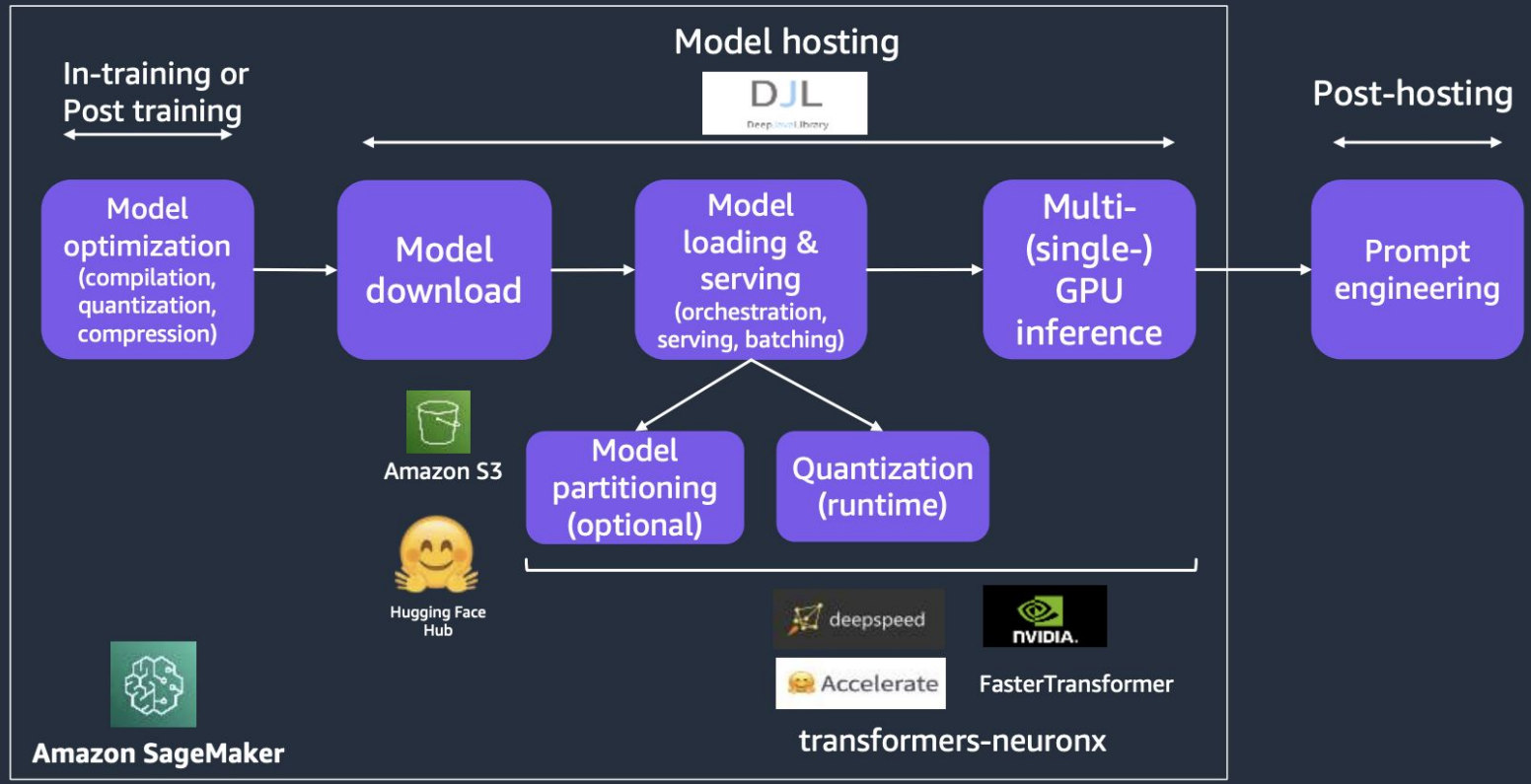
Build, train, tune, and host your own models



Selection of pre-trained models to suit your use case



Large model optimization and hosting pipeline on SageMaker



SageMaker Cost

https://medium.com/@tianjiaoliu2012/mastering-cost-efficiency-with-aws-sagemaker-a-guide-to-saving-on-machine-learning-expenses-e51fe8038c7a

1. **Choose the Right Instance Type**
2. **Stop and delete endpoints**
3. **Use Sagemaker studio when needed**
4. **Employ spot instances**
5. **Predictive scaling**
6. **Use Sagemaker BYO**
7. **Review and Adjust resource limits**

Why Sagemaker over Bedrock

1. Bedrock offers a **limited list of models**, and you can't bring your own. It's **not available in the London** region.
2. While you can fine-tune the AWS Titan model, **access to the weights is restricted (cannot fine tune)**. This causes **vendor lock-in,** so its more advantageous to use open-source LLM space using Sagemaker or EC2.
3. For prototyping with **sensitive data within the private VPC and for a production-ready model, t**he preference would be to use EC2 or SageMaker to avoid potential reengineering challenges later.
4. With SageMaker Training or Processing Jobs, it's straightforward to **bring your container from ECR** and initiate the job with a single API call (CreateTrainingJob, CreateProcessJob).
5. SageMaker JumpStart offers flexibility for **integrating Retrieval-Augmented Generation (RAG)** models based on project needs. In contrast, Bedrock's integration with RAG models depends on their availability within the Bedrock ecosystem.

Textract

https://aws-samples.github.io/amazon-textract-textractor/
https://docs.aws.amazon.com/textract/latest/dg/what-is.html
https://aws.amazon.com/textract/faqs/

How are documents processed today

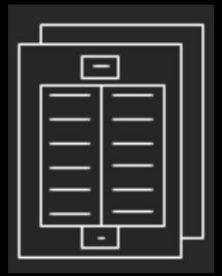
Methods

1. Manual processing
2. OCR
3. Rule and Template based extraction

Challenges

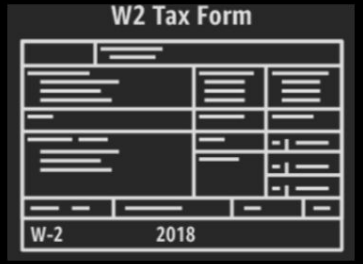
1. Can process simple docs only
2. Error prone
3. Flat bag of words
4. Cant detect
   a. Multi column
   b. Rotated text
   c. Stylish text
5. OCR reads left to right missing table structure

Amazon Textract features



Text extraction

Table extraction



Form extraction

Amazon Textract: Text extraction

Document

Herman Melville

is washed by waves, and cooled by breezes, which a few hours previous were out of sight of land. Look at the crowds of water-gazers there.

Circumambulate the city of a dreamy Sabbath afternoon. Go from Corlears Hook to Coenties Slip, and from thence, by Whitehall, northward. What do you see?—Posted like silent sentinels all around the town, stand thousands upon thousands of mortal men fixed in ocean reveries. Some leaning against the spiles; some seated upon the pier-heads; some looking over the bulwarks of ships from China; some high aloft in the rigging, as if striving to get a still better seaward peep. But these are all landsmen; of week days pent up in lath and plaster—tied to counters, nailed to benches, clinched to desks. How then is this? Are the green fields gone? What do they here?

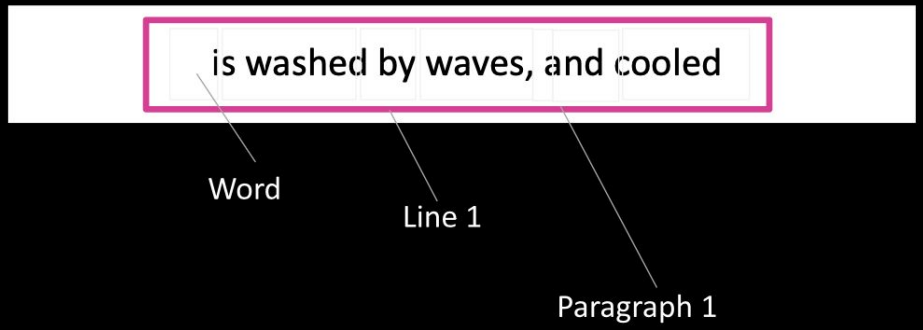
But look! here come more crowds, pacing straight for the water, and seemingly bound for a dive. Strange! Nothing will content them but the extremest limit of the land; loitering under the shady lee of yonder warehouses will not suffice. No. They must get just as nigh the water as they possibly can without falling in. And there they stand—miles of them—leagues. Inlanders all, they come from lanes and alleys, streets and avenues—north, east, south, and west. Yet here they all unite. Tell me, does the magnetic virtue of the needles of the compasses of all those ships attract them thither?

Once more. Say you are in the country; in some high land of lakes. Take almost any path you please, and ten to one it carries you down in a dale, and leaves you there by a pool in the stream. There is magic in it. Let the most absent-minded of men be plunged in his deepest reveries—stand that man on his legs, set his feet a-going, and he will infallibly lead you to water, if water there be in all that region. Should you ever be athirst in the great American

2

Output

Blocks: PAGE, PARAGRAPH, LINE, WORD



Amazon Textract: Table extraction simplified

Previous Employment History				
Start Date	**End Date**	**Employer Name**	**Position Held**	**Reason for leaving**
1/15/2009	6/30/2013	Any Company	Head Baker	Family relocated
8/15/2013	present	Example Corp.	Baker	N/A, current employer

- Table recognized
- Words grouped by cell

Output

{

Start Date: 1/15/2009

End Date: 6/30/2013

Employer Name: Any Company

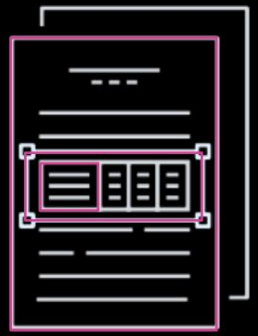
Position Held: Head Baker

Reason for leaving: Family relocated

}

Amazon Textract: Table extraction

Document



Output

Blocks: PAGE, TABLE, CELL

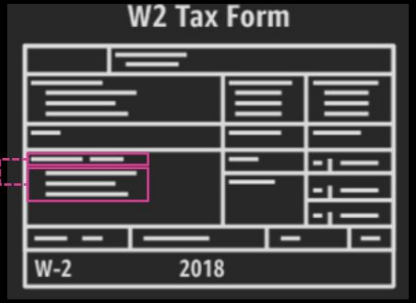
For each block, you get

- Text
- Confidence score
- Block relationships (e.g., cells within a table)

A confidence score is a number between 0 and 100 that indicates the probability that a given prediction is correct. With Amazon Textract, all extracted printed text, handwriting, and structured data are returned with bounding box coordinates, which is a rectangular frame that fully encompasses each piece of data identified. This allows you to identify the score for each extracted entity so that you can make informed decisions on how you want to use the results.

Amazon Textract: Form extraction

Document



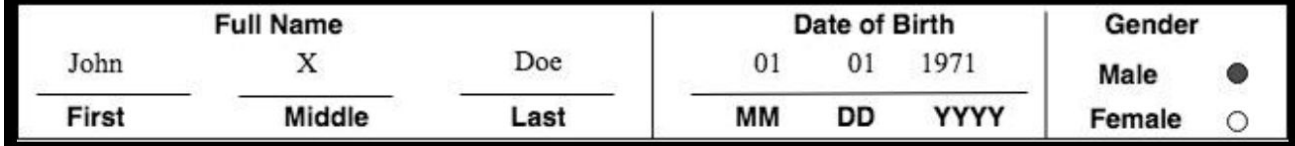
Output

Blocks: PAGE, KEY_VALUE_SET

For each block of your document

- Form field name (key) and field value (value) association
- Confidence score
- Page number
- Block relationships

Amazon Textract: Form extraction simplified



Full Name			Date of Birth			Gender	
John	X	Doe	01	01	1971	Male	●
First	Middle	Last	MM	DD	YYYY	Female	○

Output

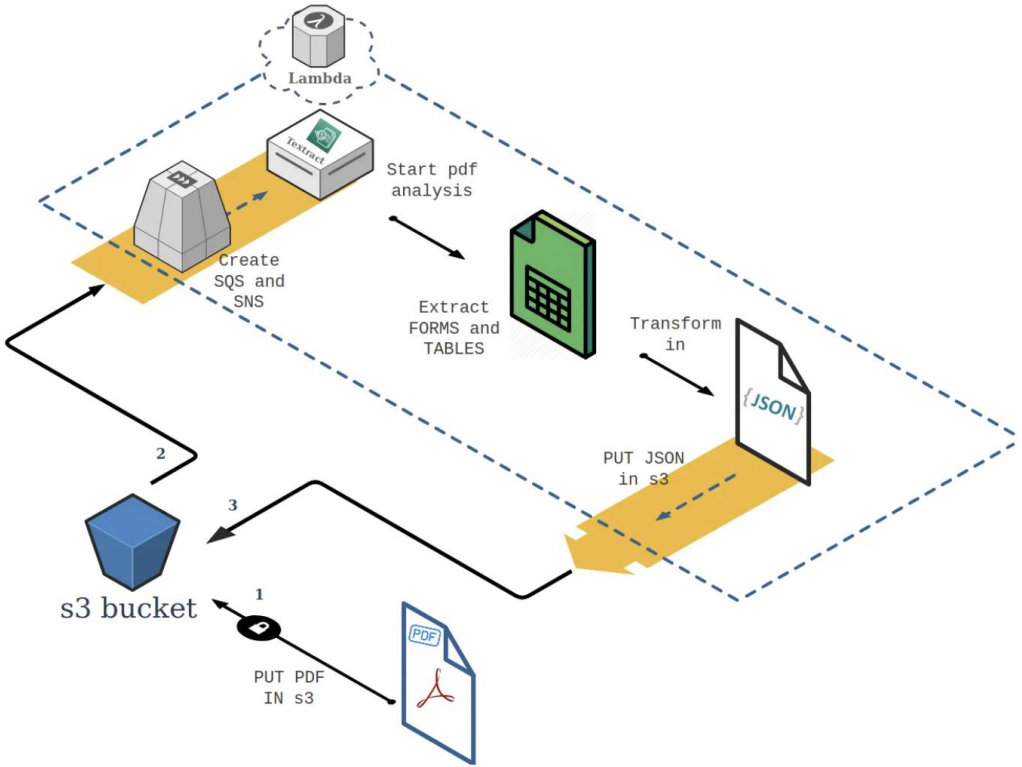
Full Name:
- First: John
- Middle: X
- Last: Doe

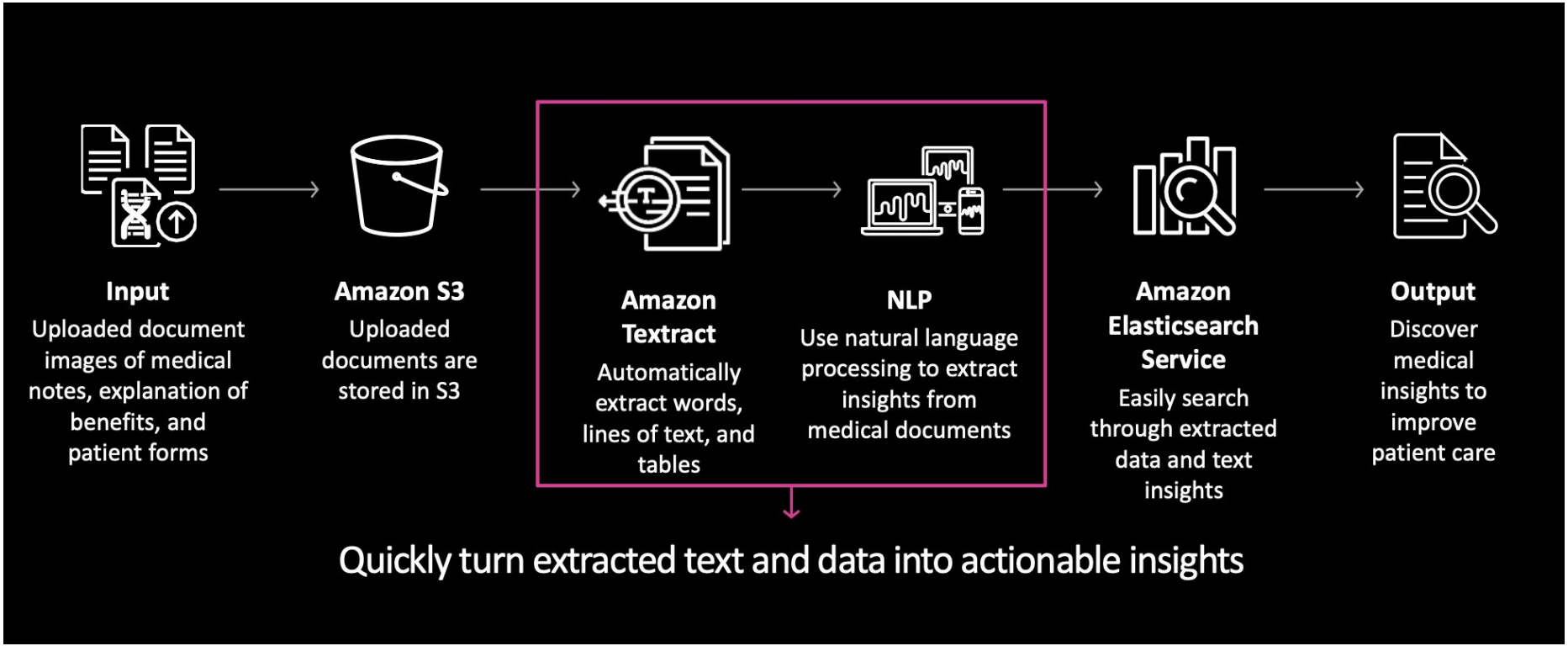
Date of Birth:
- MM: 01
- DD: 01
- YYYY: 1971

Gender:
- Male: True
- Female: False

- Logical groupings captured
- Relationships captured
- Glyphs captured

Architecture suggestion




Input
Uploaded document images of medical notes, explanation of benefits, and patient forms
Amazon S3
Uploaded documents are stored in S3
Amazon Textract
Automatically extract words, lines of text, and tables
NLP
Use natural language processing to extract insights from medical documents
Amazon Elasticsearch Service
Easily search through extracted data and text insights
Output
Discover medical insights to improve patient care
Quickly turn extracted text and data into actionable insights

Haystack x Mixtral87B

Haystack x Mixtral87B

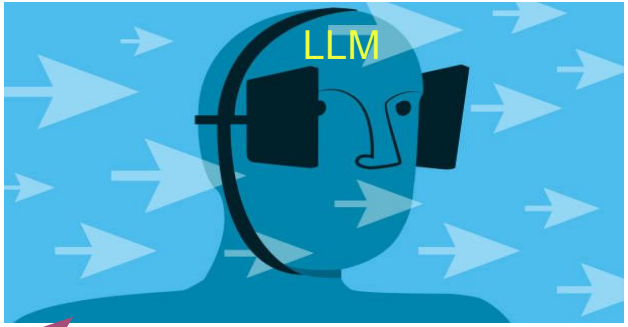
RAG (Retrieval augmented generation)

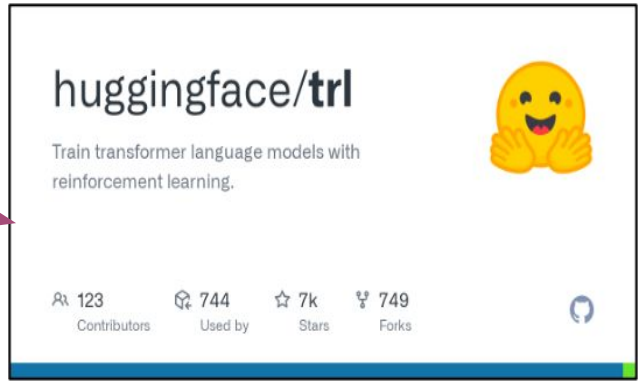
The idea behind RAG (Retrieval augmented generation) is simple: LLMs do not know the entire world, least of all your specific world.

But, with the use of retrieval techniques, RAG can provide the most useful pieces of information to an LLM so that it has the context with which to reply to queries that it otherwise would not have been trained to know about or answer.

Where could this be useful?

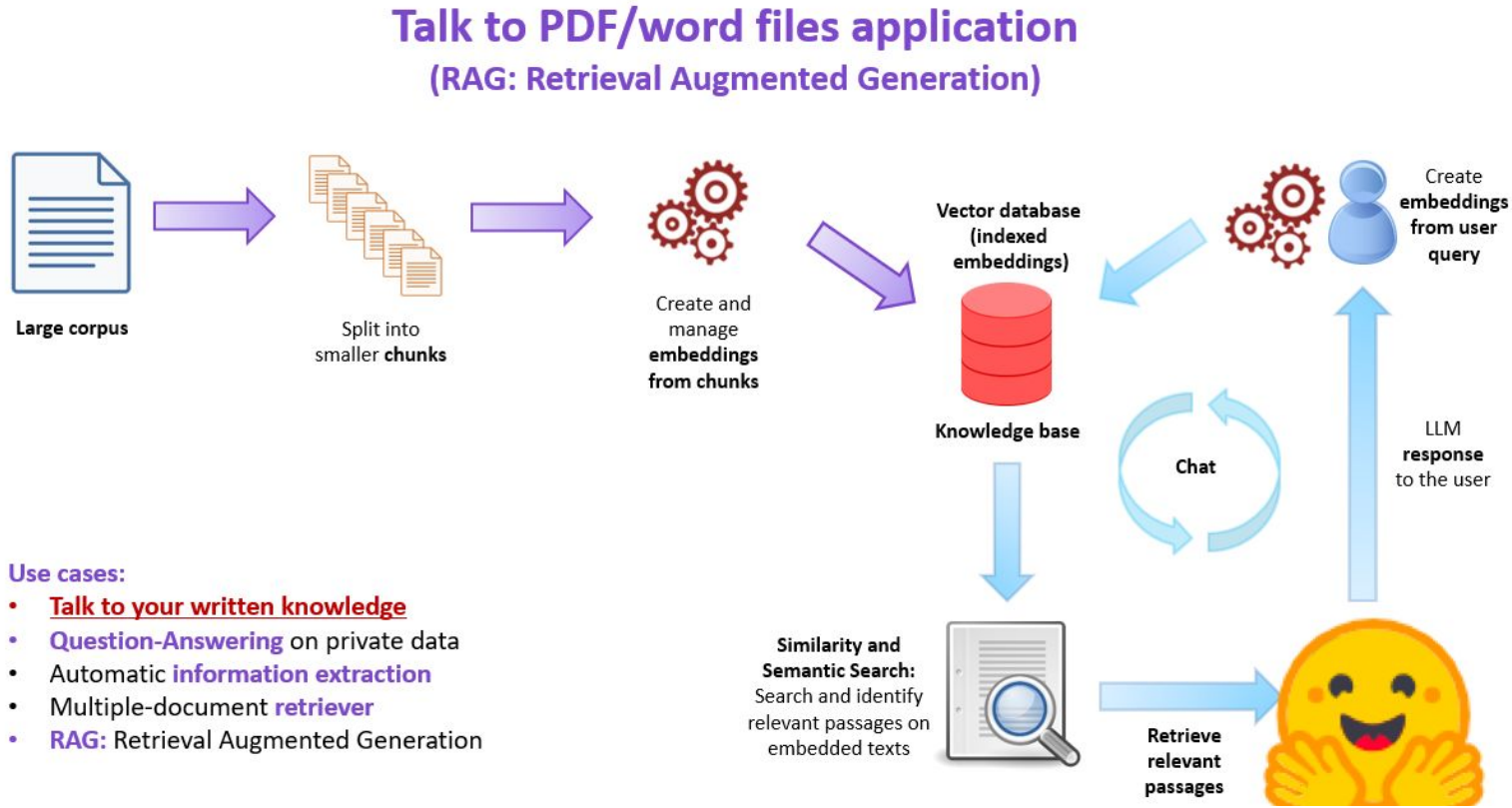
1) Data that are internal or very specialised and you know that an LLM training set will not probably have included them.
2) When you want to apply “guardrails” to the LLM that you are using. Prefer a “answer not known” response to an **LLM “hallucinating**”
3) you can always get the latest information/answer around a topic or question without worrying about when was the training cut off for the foundation model.
4) You can avoid spending time, effort and money on complex process of fine tuning or eventually training on your data.





NOTE: For more serious attempts of finetuning of LLMs the usage of dedicated libraries such as **trl** is recommended together with the **usage of compute & GPUS**

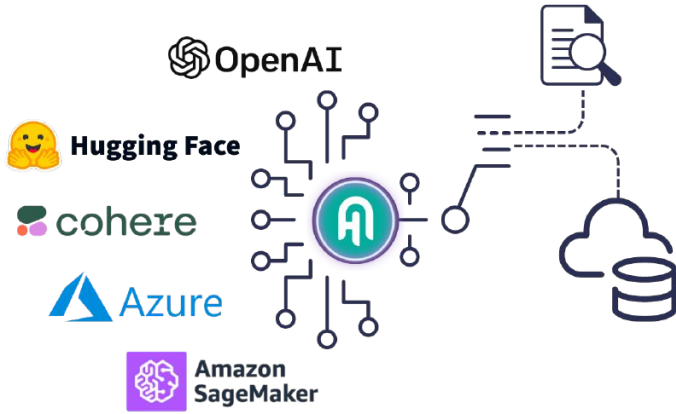
RAG Example


Talk to PDF/word files application
(RAG: Retrieval Augmented Generation)
Large corpus
Split into smaller chunks
Create and manage embeddings from chunks
Vector database (indexed embeddings)
Knowledge base
Create embeddings from user query
Chat
LLM response to the user
Similarity and Semantic Search: Search and identify relevant passages on embedded texts
Retrieve relevant passages
Use cases:
Talk to your written knowledge
Question-Answering on private data
Automatic information extraction
Multiple-document retriever
RAG: Retrieval Augmented Generation

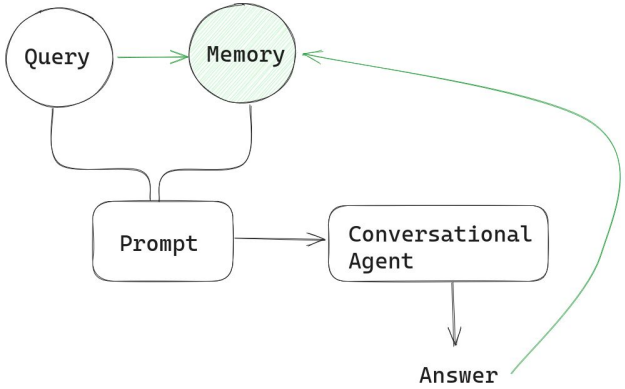
What is Haystack?

Haystack is an LLM app orchestrator (think of it as Airflow for LLMs)

- Haystack is built around the concept of pipelines. A pipeline is a powerful structure that performs an NLP task. It's made up of components connected together.
- Ready made components and integrations.
  - API connectors : $$: OpenAI,Cohere etc & Open Source : (Hugging Face models)
  - Document Stores : In Memory, MySQL, Postgres, Elasticsearch, Opensearch, Pinecone
  - Preprocessing Tools
  - Components to help with retrieval:
    - on documents (BM25),
    - embeddings (cosine distance)
  - Notebook integrations (Collab,Jupyter,AWS Sagemaker)
  - REST API endpoints

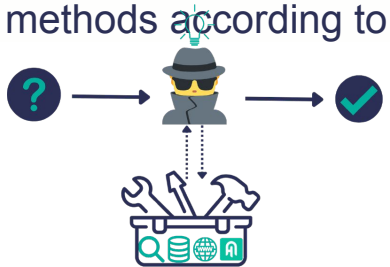


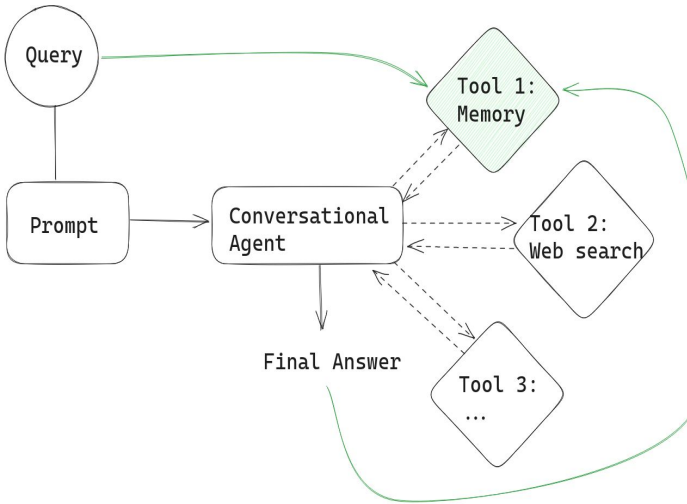
What is Haystack? pt2



Haystack is an LLM orchestrator that can also help with :

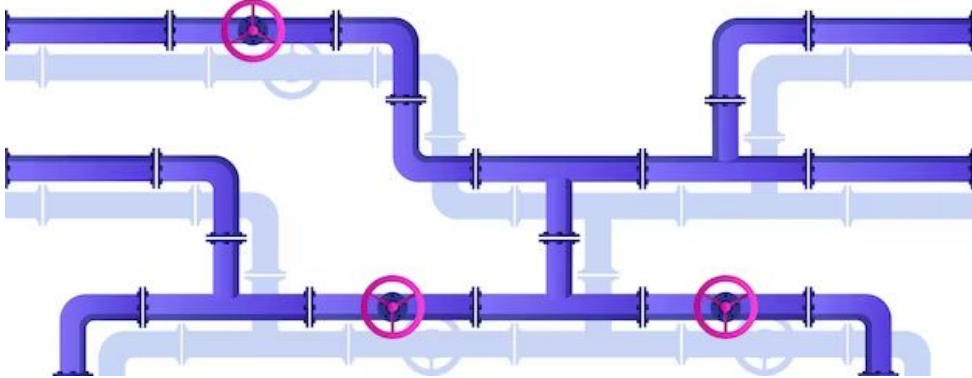
- Providing **memory** to the conversation/interaction
  - By default, LLMs don't have an inbuilt notion of memory. As far as an out-of-the-box LLM is concerned, every prompt it receives is the beginning of an entirely new interaction.
  - Haystack provides an easy way to have user discussion memory as part of a pipeline
- Providing a way to selectively or autonomously use different pipelines or retrieval methods according to tasks via **Agents**





What is Haystack? pt3

Haystack pipeline components in detail



Data Handling

Crawler
DocumentClassifier
DocumentLanguageClassifier
EntityExtractor
FileClassifier
FileConverter
PreProcessor

Semantic Search

Ranker
Reader
Retriever
SearchEngine
QuestionGenerator

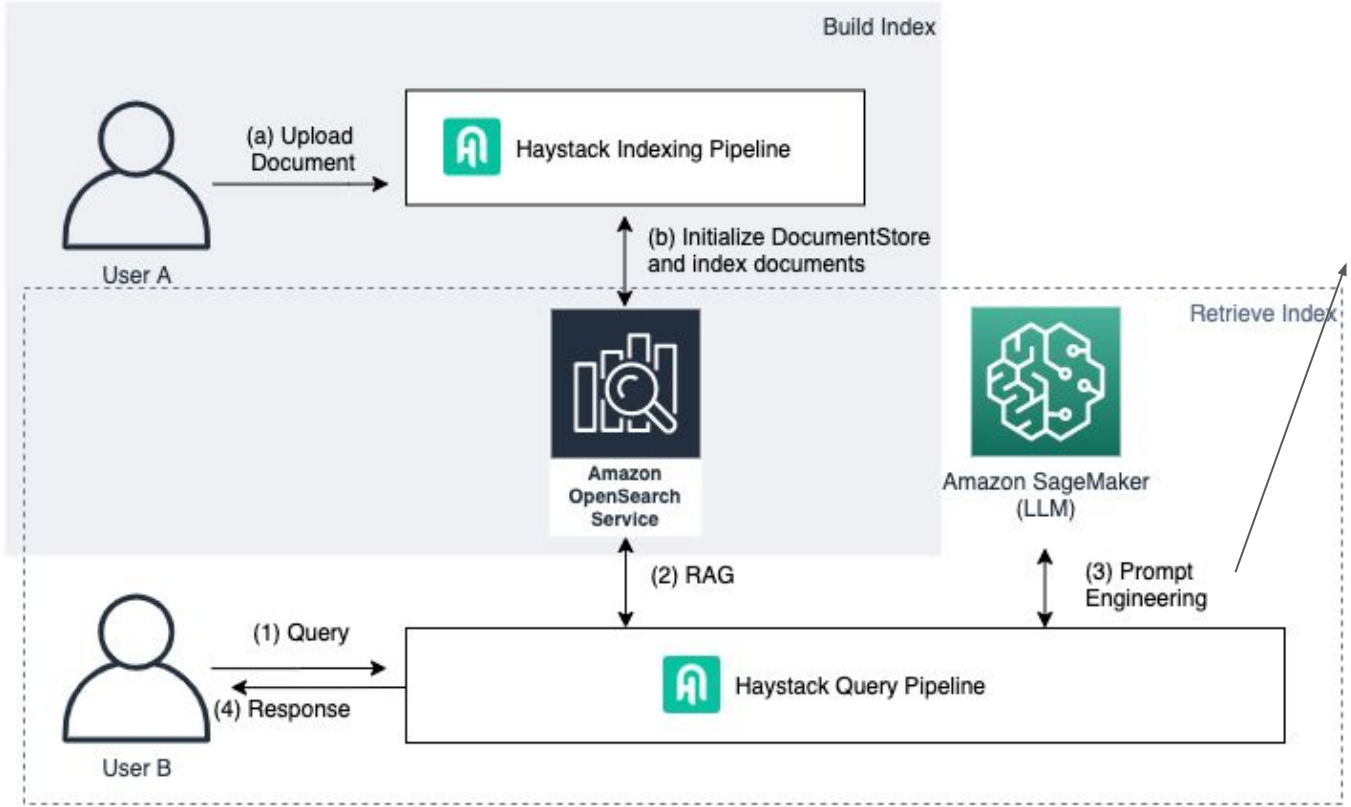
Routing

QueryClassifier
RouteDocuments

Utility Components

DocumentMerger
Docs2Answers
JoinAnswers
JoinDocuments
PseudoLabelGenerator
Shaper
Summarizer
TransformersImageToText
Translator
WhisperTranscriber

Retrieve Document Insight with Haystack, Amazon OpenSearch and Amazon SageMaker




```
qa_template =

PromptTemplate(prompt=
 """ Using the information
contained in the context,
answer only the question asked
without adding question
suggestions

 If the answer cannot be
inferred from the context,
reply: '\I don't know'.
 Context: {join(documents)};
 Question: {query}
 """)
```

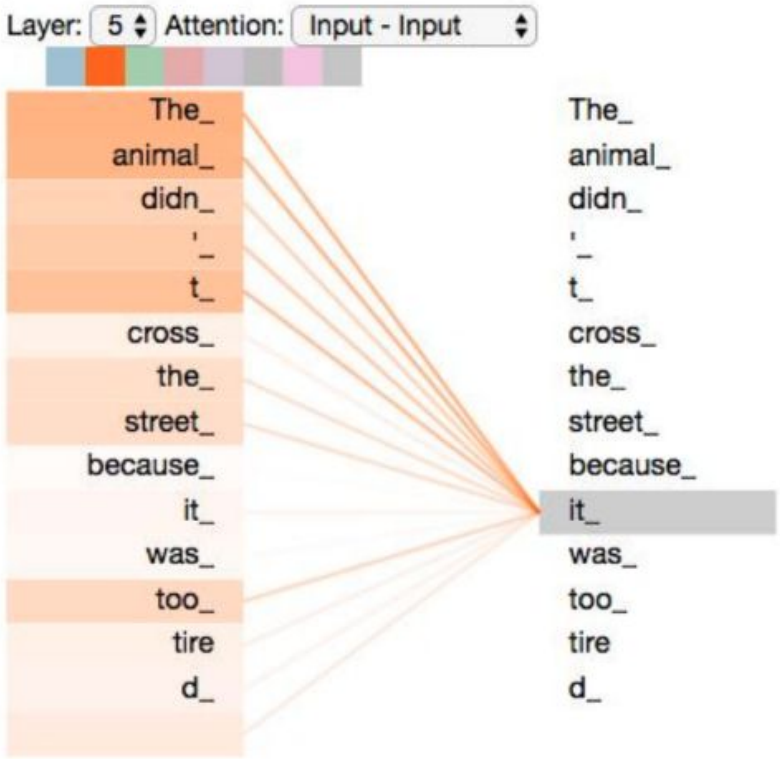
Fundamentals of Prompt Engineering

Why fundamentals?

Context window limitations

Attention mechanism

Model name	Window
flan-t5	512
text-davinci-003	4,097
gpt-4	8,192 32,768



Fundamentals of Prompt Engineering

```
[28] pn = PromptNode("text-davinci-003", api_key=openai_key)
     poem = pn("Write a short poem about Haystack")
     print(poem[0])
```

```
Rock

A rock of grandeur stands tall in the sea,
A landmark admired by all, one can see.
It stands as a symbol of nature's might,
A beacon of beauty, Haystack Rock's light.
```

```
[29] prompt = """
     "Write a short poem about Haystack,
     an open-source NLP framework,
     focus on recently released Haystack PromptNode and its powerful templates"""
     poem = pn(prompt)
     print(poem[0])
```

```
Haystack PromptNode is here
It brings new tools and templates without fear
Enabling NLP task automation with ease
Haystack's power is clear!
```

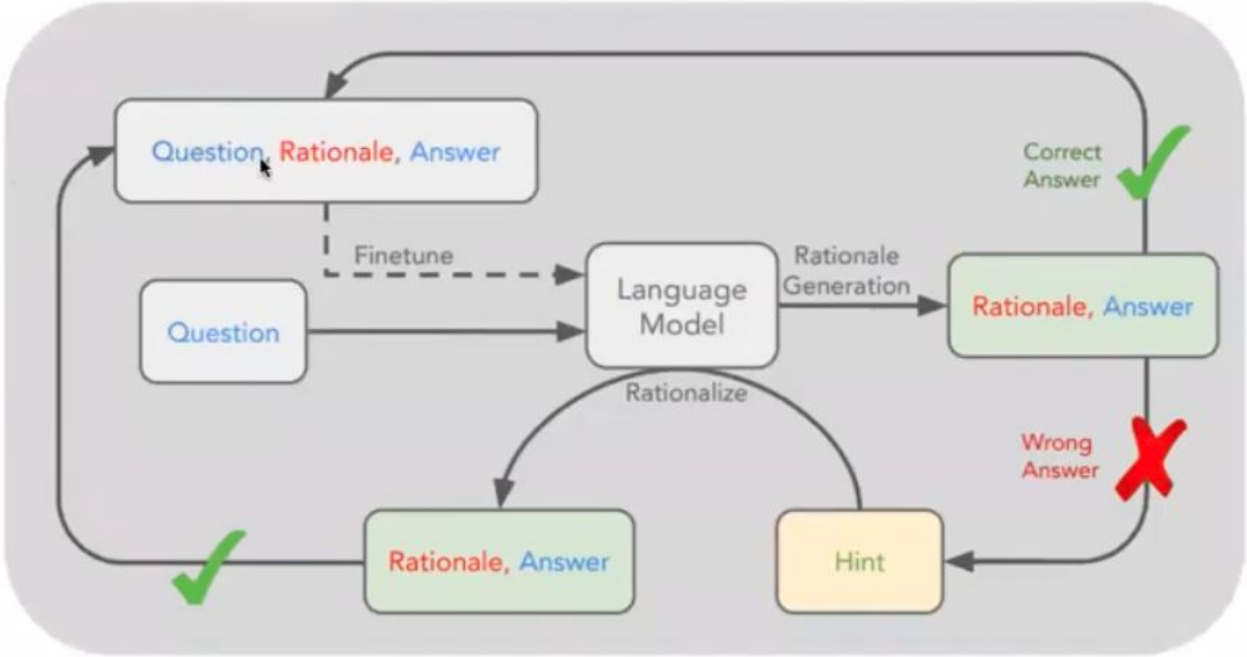
Clarity and specificity

Prompt Engineering Guidelines

- Provide clear and explicit instructions
- Include examples if needed
- Experiment with different prompt styles
- Utilize system-level constraints (temperature, token limits)
- Iterate and refine prompts based on feedback
- Address potential biases in LLMs and prompts

Prompting and latest research

Self-Taught Reasoner (STaR) method



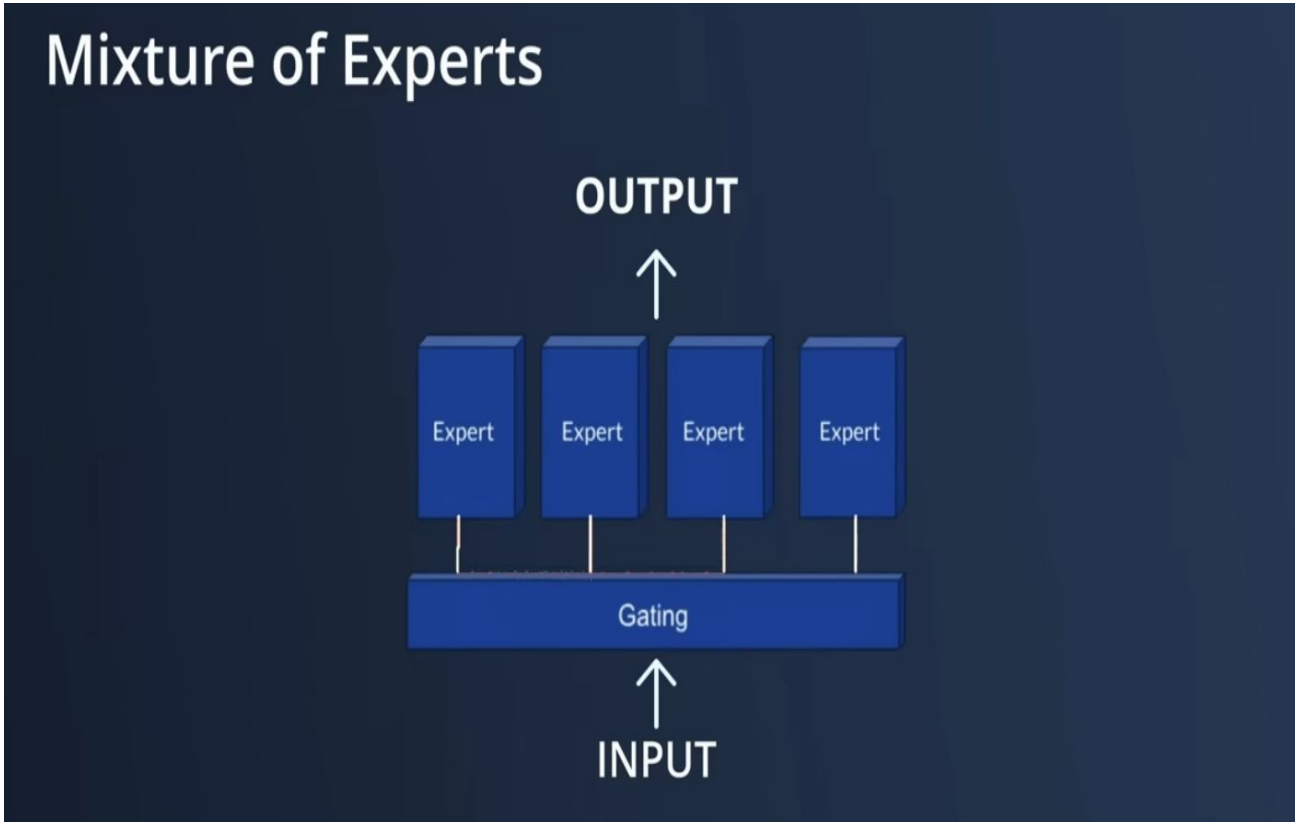
MIXTRAL87B LLM model

MIXTRAL87B LLM model

https://www.youtube.com/watch?v=RYZ0FMAKRFs

- It takes 8 separate 7 billion parameter models which are experts in different things and puts together into single model
- In normal network we have one input and pass on to expert and get one output
- It can allocate to multiple experts as well
- We can pretrain one expert more specifically
- **GPT4 is (MOE) - 8 Experts and each 220 billion parameters**

MIXTRAL87B LLM model



4 experts and input goes to gating layer and it decide which expert to allocate to this prompt and sends signal to that expert and it sends output.

Allows expert to specialise in different tasks

Gating is trained to get better able to determine what the input is and which expert it should allocate task to

MIXTRAL87B LLM model

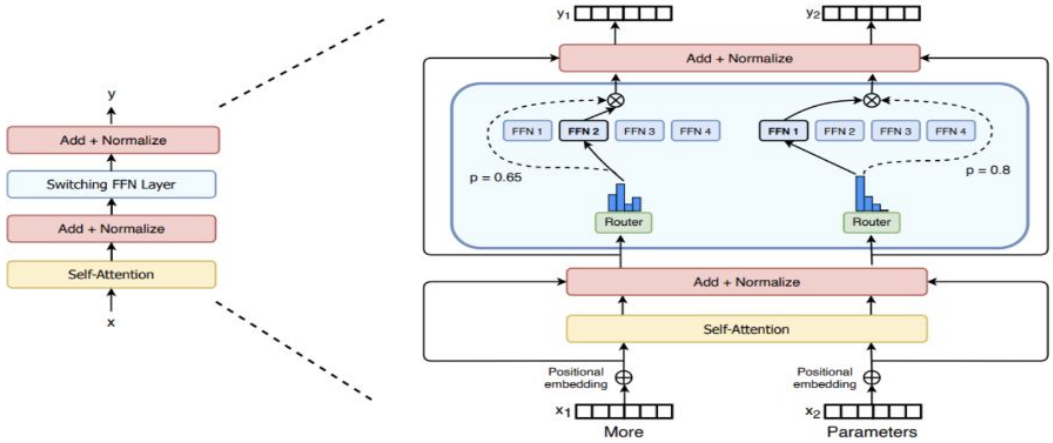


Figure 2: Illustration of a Switch Transformer encoder block. We replace the dense feed forward network (FFN) layer present in the Transformer with a sparse Switch FFN layer (light blue). The layer operates independently on the tokens in the sequence. We diagram two tokens (x_1 = "More" and x_2 = "Parameters" below) being routed (solid lines) across four FFN experts, where the router independently routes each token. The switch FFN layer returns the output of the selected FFN multiplied by the router gate value (dotted-line).

Why MIXTRAL87B LLM model?

- Opensource
- Mixtral-8x7B is Mistral AI's second Large Language Model (LLM). circa 45 Billion params
- It is the strongest **open-weight** model with a permissive license and the best model overall regarding cost/performance trade-offs. In particular, it **matches** or outperforms **GPT3.5** on most standard benchmarks.
- Faster because it uses specific experts(?)
- There is also mixtral medium model (paid version)

https://huggingface.co/blog/mixtral

HF chatbot-arena-leaderboard

Model	Arena Elo rating
GPT-4-Turbo	1233
GPT-4-0314	1191
GPT-4-0613	1157
Claude-1	1151
Claude-2.0	1130
Claude-2.1	1120
GPT-3.5-Turbo-0613	1116
Mixtral-8x7b-Instruct-v0.1	1116
Claude-Instant-1	1110
Tulu-2-DPO-70B	1110
Yi-34B-Chat	1109
Gemini Pro	1106
GPT-3.5-Turbo-0314	1105

DEMO

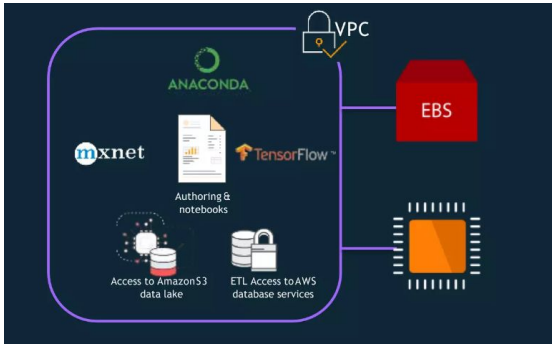
Use cases

1. Webscrape - AP (Prompt Node + Q&A)
2. Webscrape - Cloudplatform (Prompt Node + Q&A)
3. Video -> text + Q & A

Questions?

LLM on internal data applications

1. Access HuggingFace's private endpoints at https://huggingface.co/inference-endpoints for $6 per hour, ensuring datasets stay within our VPC.
2. Deploy your own Sagemaker LLM instance (g5.45xlarge 💰) (https://www.philschmid.de/sagemaker-deploy-mixtral)



4. Finetune* existing LLMs to then later deploy using AWS Trainium ec2 instances:

instance size	accelerators	accelerator memory	vCPU	CPU Memory	price per hour
trn1.2xlarge	1	32	8	32	$1.34
trn1.32xlarge	16	512	128	512	$21.50
trn1n.32xlarge (2x bandwidth)	16	512	128	512	$24.78

LLM on internal data applications : finetuning etc (if time permits)

🤗 PEFT: Parameter-Efficient Fine-Tuning of Billion-Scale Models on Low-Resource Hardware

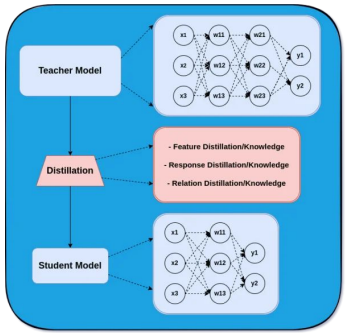
- **LoRA** can **reduce** the number of trainable parameters by **10,000** times and the **GPU memory requirement by 3 times**. LoRA performs on-par or better than fine-tuning in model quality on RoBERTa, DeBERTa, GPT-2, and GPT-3, despite having fewer trainable parameters (https://arxiv.org/abs/2106.09685) . Also see **sLora** (https://arxiv.org/abs/2308.06522)

🤗 Quantization techniques reduce memory and computational costs by representing weights and activations with lower-precision data types like 8-bit integers (int8) or 4-bit etc . This enables loading larger models you normally wouldn't be able to fit into memory, and speeding up inference.

Model	Average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
mistralai/Mixtral-8x7B-v0.1	68.42	66.04	86.49	71.82	46.78	81.93	57.47
TheBloke/Mixtral-8x7B-v0.1-GPTQ	65.7	65.19	84.72	69.43	45.42	81.14	48.29
Score Delta	**0.960**	**0.987**	**0.980**	**0.967**	**0.971**	**0.990**	**0.840**

🤗 Distillation

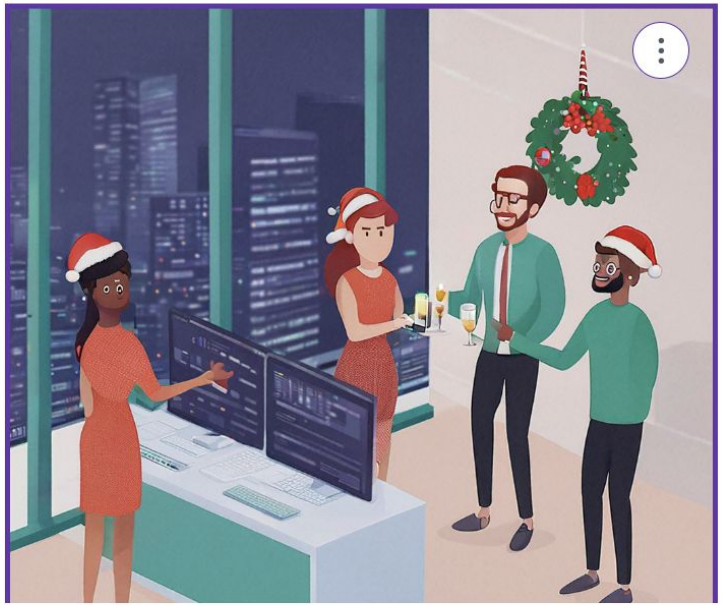
Distillation involves using the output from a larger, more capable 'teacher' model to train a smaller 'student' model, enhancing its accuracy.



Knowledge Distillation

Titan Image Generator G1 *v1* | ODT

Change



cartoon Data engineers for data platform team at christmas party

Run

Configurations

Reset

Mode Info

Generate | Edit

Negative Prompt Info

Add negative prompt

Reference Image Info

Upload image

Response Image Info

Quality

Standard

Tasks Libraries Datasets Languages Licenses Other

Filter Tasks by name

Multimodal

- Feature Extraction
- Text-to-Image
- Image-to-Text
- Image-to-Video
- Text-to-Video
- Visual Question Answering
- Document Question Answering
- Graph Machine Learning
- Text-to-3D
- Image-to-3D

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Image-to-Image
- Unconditional Image Generation
- Video Classification
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Conversational
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity

Audio

- Text-to-Speech
- Text-to-Audio

Models 453,649

Filter by name

new Full-text search

↑↓ Sort: Trending

TinyLlama/TinyLlama-1.1B-Chat-v1.0
Text Generation • Updated 1 day ago • ↓5.08k • ♡ 274

microsoft/phi-2
Text Generation • Updated 20 days ago • ↓117k • ♡ 1.65k

h94/IP-Adapter-FaceID
Text-to-Image • Updated about 8 hours ago • ↓26.4k • ♡ 308

cognitivecomputations/dolphin-2.5-mixtral-8x7b
Text Generation • Updated 2 days ago • ↓24.5k • ♡ 856

cognitivecomputations/dolphin-2.6-mistral-7b
Text Generation • Updated 5 days ago • ↓1.02k • ♡ 71

mistralai/Mistral-7B-v0.1
Text Generation • Updated 23 days ago • ↓435k • ♡ 2.48k

meta-llama/Llama-2-7b-chat-hf
Text Generation • Updated Nov 13, 2023 • ↓659k • ♡ 2.32k

NousResearch/Nous-Hermes-2-Yi-34B
Text Generation • Updated 1 day ago • ↓2.99k • ♡ 115

NousResearch/Nous-Hermes-2-SOLAR-10.7B
Text Generation • Updated about 11 hours ago • ↓55 • ♡ 46

stabilityai/stable-video-diffusion-img2vid-xt
Image-to-Video • Updated Dec 1, 2023 • ↓93k • ♡ 1.5k

mistralai/Mixtral-8x7B-Instruct-v0.1
Text Generation • Updated 19 days ago • ↓392k • ♡ 1.65k

dataautogpt3/OpenDalleV1.1
Text-to-Image • Updated about 22 hours ago • ↓47.9k • ♡ 300

argilla/notux-8x7b-v1
Text Generation • Updated 6 days ago • ↓1.04k • ♡ 115

stabilityai/stable-diffusion-xl-base-1.0
Text-to-Image • Updated Oct 30, 2023 • ↓4.79M • ♡ 4.01k

upstage/SOLAR-10.7B-Instruct-v1.0
Text Generation • Updated 6 days ago • ↓15.1k • ♡ 397

stabilityai/sdxl-turbo
Text-to-Image • Updated 27 days ago • ↓606k • ♡ 1.43k

mistralai/Mixtral-8x7B-v0.1
Text Generation • Updated 19 days ago • ↓86.3k • ♡ 862

TinyLlama/TinyLlama-1.1B-intermediate-step-1431k-3T
Text Generation • Updated 4 days ago • ↓2.29k • ♡ 46

mistralai/Mistral-7B-Instruct-v0.2
Text Generation • Updated 19 days ago • ↓128k • ♡ 529

upstage/SOLAR-10.7B-v1.0
Text Generation • Updated 6 days ago • ↓6.28k • ♡ 138

Playground
Assistants
Fine-tuning
API keys
Files
Usage
Settings

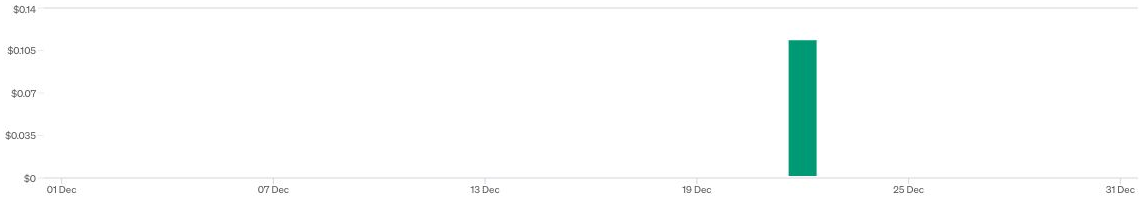
Usage

Cost | Activity

December | Export

Daily Costs $0.11

Audio models $0.11



Monthly Bill Dec 1 - 31

$0.11

/ $120.00 limit

Increase limit

Credit Grants USD

Used | Expired

$0.12 / $28.00

Invoices

Dec 2023	Paid	$12.00

Documentation
Help
All products