



# ChatGPT

insights, uses, and impacts

**Michele FILANNINO**

Principal Data Scientist

MODENA, 03 MARCH 2023



**UNIMORE**  
UNIVERSITÀ DEGLI STUDI DI  
MODENA E REGGIO EMILIA



We are a **consulting, software development** and **economic research** company serving banks, insurance companies and enterprises. We specialize in risk, wealth and asset management services and financial consulting.

**1100+** **500+**  
PEOPLE CLIENTS

#### BRANCHES

- Istanbul
- Cairo
- London
- Moscow
- Zurich

#### OFFICES

- Bologna
- Milan
- Rome

#### PARTNERSHIPS

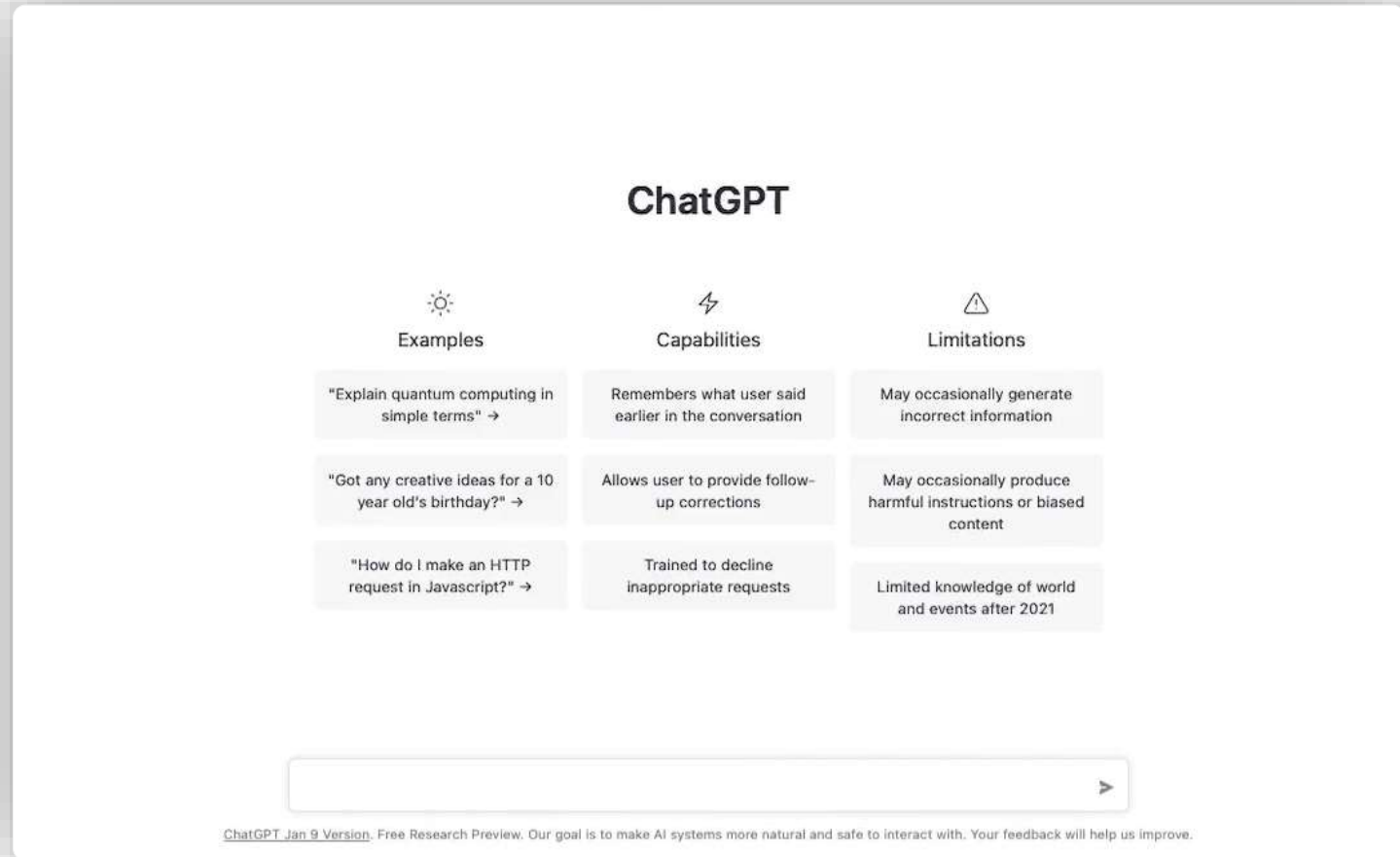
- Amman
- Athens
- Dubai
- Kuwait City
- Lagos
- Lomè
- Madrid
- Paris
- Stockholm
- Tunis
- Vienna



# ChatGPT: what it is

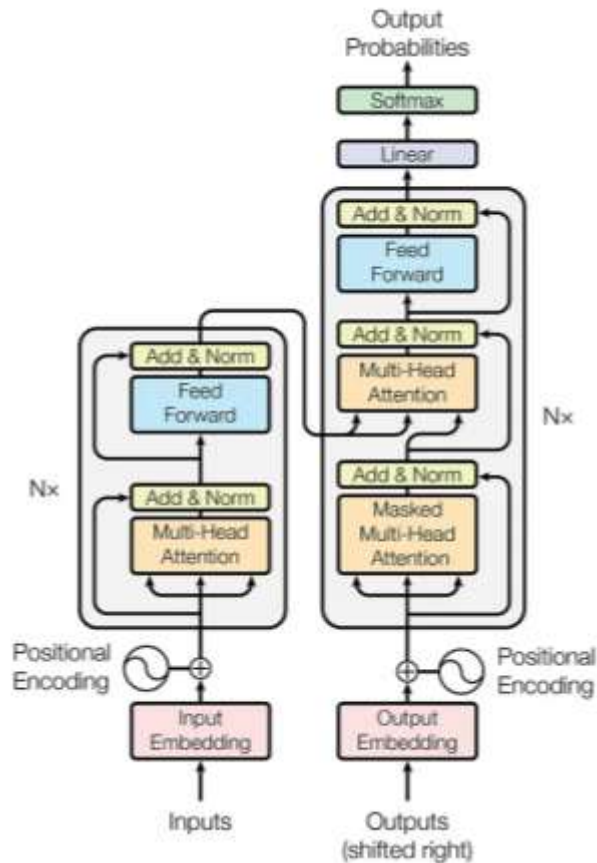
natural language processing ♥ machine learning

ChatGPT is a **chatbot** designed to understand and generate human-like language. It is based on the GPT **(Generative Pre-training Transformer)** language model developed by **OpenAI**, which has been trained on a large dataset of human language in order to better understand how humans communicate.



# transformers

where the revolution started



Introduced in 2017 by researchers from Google Brain to overcome limitations of the previous best neural network architecture:

- Long-Short Term Memories (LSTMs).

They were slow to train and not really bi-directional. The transformer addresses such problems:

- it's **faster to train** (words can be processed simultaneously),
- and **learns deeper context** from words bi-directionally.

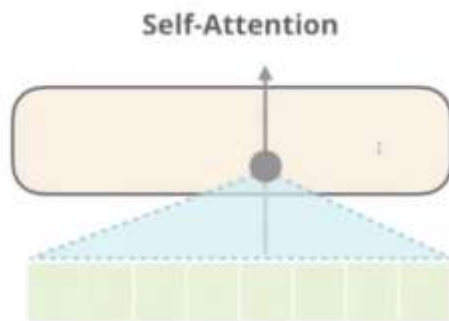
# (large) language models

size does matter

## BERT/RoBERTa/ALBERT

bi-directional encoders

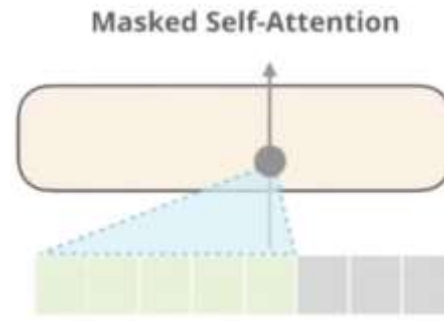
“you give it a piece of text, it'll look left and right, right and left and it will encode the meaning of each word”.



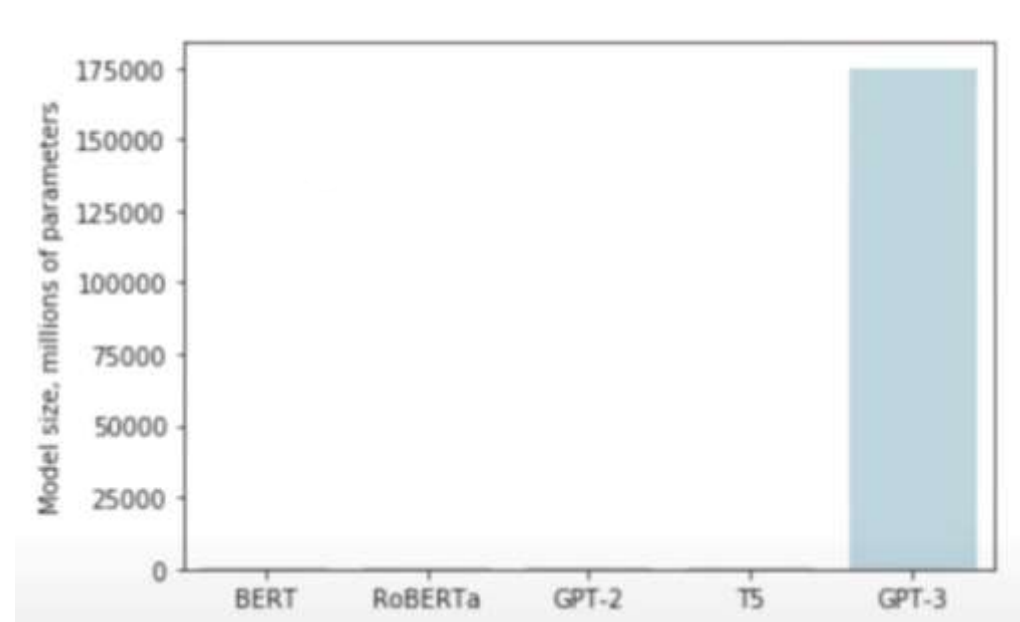
## GPT/XLNET

autoregressive decoders

“you give it a piece of text (prompt), it'll try to guess the next words”



“Each **increase [in size]** has brought improvements in **text synthesis and/or downstream NLP tasks**, and there is evidence suggesting that log loss, which correlates well with many downstream tasks, follows a smooth trend of improvement with scale.”





# copywriting

## blockchain example

---

### **MF** **Scrivi un testo di 500 parole sulla blockchain**

ChatGPT La blockchain è una tecnologia che permette di creare un registro digitale distribuito e immutabile. In pratica, la blockchain è una lista di blocchi di dati che vengono concatenati tra loro in modo tale da formare una catena. Ogni blocco contiene informazioni, come ad esempio delle transazioni economiche o dei dati personali, e una serie di metadati, come ad esempio l'ora in cui il blocco è stato creato e il suo indirizzo nella catena. La caratteristica principale della blockchain è che essa è decentralizzata, ovvero non dipende da alcuna autorità centrale che ne gestisca l'accesso o l'utilizzo. Invece, la blockchain viene gestita da una rete di nodi che collaborano tra loro per validare e aggiungere nuovi blocchi alla catena. Ogni nodo contiene una copia dell'intera blockchain e, per questo motivo, la blockchain è considerata un sistema altamente sicuro e affidabile.

Una delle prime e più famose applicazioni della blockchain è stata quella di creare una criptovaluta, ovvero il Bitcoin. Il Bitcoin è una valuta digitale che viene creata e scambiata utilizzando la blockchain come supporto. Grazie alla decentralizzazione ...

# features & capabilities

it somehow knows the styles



Ciao ChatGPT! Mi aiuti a trovare  
che non è corretto mandare rich



Ciao! Ecco alcune possibili opzio  
appropriato inviare richieste di r

1. "Gentile [nome], per rispetta  
chiedo cortesemente di evita  
lavorativo. Grazie per la comp
2. "Ciao [nome], ti prego di ten  
concentrarmi sulla mia vita p  
riunioni in questi momenti."
3. "Salve [nome], per favore evi  
lavorativo. Mi aiuterà a mante



Ma invece se fosse Barack Obama a scriverlo?



Barack Obama è stato anche un ottimo oratore e scrittore, quindi per scrivere una frase  
in modo simile a come lui avrebbe potuto farlo, ti consiglio di utilizzare un linguaggio  
semplice e diretto, con un tono di persuasione e di incoraggiamento. Ecco alcune  
possibili opzioni:



1. "Sappiamo tutti quanto sia importante mantenere un equilibrio tra il lavoro e la vita privata, per questo chiedo a tutti di evitare di inviare richieste di riunioni fuori dall'orario lavorativo. Insieme possiamo creare un ambiente di lavoro più sano e produttivo."
2. "La vita privata è importante, e fuori dall'orario lavorativo dobbiamo rispettare il nostro spazio e il nostro tempo. Chiedo quindi a tutti di evitare di inviare richieste di riunioni in questi momenti, perché è giusto così."
3. "Ciascuno di noi ha il diritto di avere un equilibrio tra lavoro e vita privata. Ecco perché chiedo a tutti di evitare di inviare richieste di riunioni fuori dall'orario lavorativo. Insieme possiamo creare un ambiente di lavoro più rispettoso e produttivo."

# software development

how do we use it?

- It can help you **generate code snippets** and sample implementations for specific programming tasks or languages.
- It can assist with **debugging** and troubleshooting by providing suggestions and potential solutions to coding issues.
- It can also be used to generate or assist in the creation of software **documentation**, technical reports, or other written materials related to software development.
- Additionally, we can use it to **generate test data** for your application, or to perform testing and validation of your software.
- It can assist with the **optimization of your code**, by providing suggestions for performance improvements, memory management, and other optimization techniques.
- It can help you with generating or **optimizing SQL queries**, by providing suggestions for query optimization and performance tuning.
- It can help with automating repetitive or time-consuming tasks in the development process, such as **code understanding**, code generation, **refactoring**, formatting, writing idiomatic constructs, or test cases.



**Anectode:** Andrej Karpaty, senior director of AI at Tesla, recently admitted that ~80% of his code gets automatically written by GitHub Copilot.



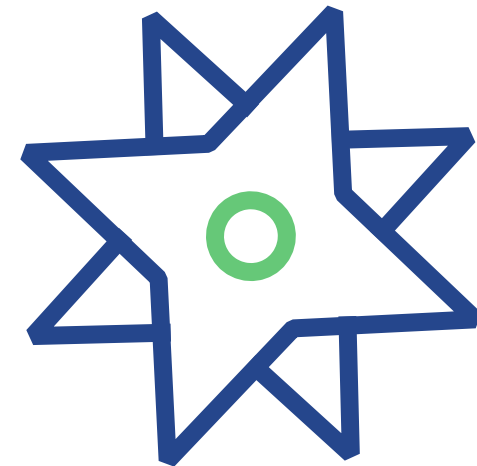


# general domain usage

how do we use it?

---

- It can help you with specific tasks or projects, such as data analysis, research, or **report generation**. I can also assist with more general problem-solving or decision-making by providing information and **insights on a particular topic**.
- It can provide automated, **real-time support for your business operations** by answering customer queries, providing product recommendations, or handling **simple data entry tasks**.
- It can be used as a **virtual assistant to help with scheduling, email management**, and other administrative tasks.
- It can be used as a tool to help you **train employees, students, or customers** on a particular topic or skill.
- It can help **generate content**, such as writing articles, **presentations**, stories, and scripts, or creating visual designs and videos.
- It can help getting ideas about novel **use cases** for specific new markets.



# market evolution

generative AI landscape: companies/projects

A brand **new** market landscape popped up almost out of the blue. Some companies specialise in **text-to-text** (e.s. OpenAI ChatGPT), **text-to-image** (OpenAI DALL·E, Midjourney, Stable Diffusion), **text-to-code** (GitHub Copilot, OpenAI Codex), and some in **audio-to-text** (OpenAI Whisper).

Text-to-Image (T2I)	DALL·E 2	Stable Diffusion	craiyon	Jasper
	Imagen	Midjourney	NightCafe	GauGAN2
	WOMBO	Wonder	pixray-text2image	neural love
Text-to-Video (T2V)	runway	Fliki	synthesia	Meta AI
Text-to-Audio (T2A)	Play.ht	MURF.ai	RESEMBLE.AI	WELLSAID
	descript			
Text-to-Text (T2T)	Simplified	Jasper	frase	EleutherAI
	Requestory			
	grammarly	copy.ai	MarketMuse	AI21labs
	HubSpot			
	InferKit	GooseAI	Research AI	Writesonic
	cohere	CHIBI		
	Ideas AI	copysmith	Flowrite	MECHES\$S
	sudo write			
	ideasbyai	text.cortex	OpenAI GPT-3	Blog Idea Generator

Text-to-Motion (T2M)	ITREE Ind.	MDM: Human Motion Diffusion Model
Text-to-Code (T2C)	replit	Generate code
	GitHub Copilot	
Text-to-NFT (T2N)	LensAI	
Text-to-3D (T2D)	DreamFusion	CLIP-Mesh
	GET3D	
Audio-to-Text (A2T)	descript	AssemblyAI
	Whisper	OpenAI
Audio-to-Audio (A2A)	AudioLM	VOICEMOD
Brain-to-Text (B2T)	speech from brain	non-invasive brain recordings
Image-to-Text (A2T)	neural love	GPT-3 x Image Captions
	OpenAI	

# ChatGPT



## Examples

"Explain quantum computing in simple terms" →

"Got any creative ideas for a 10 year old's birthday?" →

"How do I make an HTTP request in Javascript?" →



## Capabilities

Remembers what user said earlier in the conversation

Allows user to provide follow-up corrections

Trained to decline inappropriate requests



## Limitations

May occasionally generate incorrect information

May occasionally produce harmful instructions or biased content

Limited knowledge of world and events after 2021



# limitations

be careful in determining the appropriate level of human oversight

- it **doesn't learn by the interaction** (both facts and communication style)
- at the moment, there isn't a way to make him learn from new data
- it's **not multi-modal** (text only: no images, videos, documents)
- it **can't access the Internet** for its users
- it has **limited proprietary knowledge** of the world and events after 2021



Chi è il presidente del consiglio Italiano?



Attualmente il presidente del consiglio italiano è Mario Draghi.

- it can't say **where** it did pick the information from (lack of references)
- it suffers from a **Dunning-Kruger effect**\*



Hai presente il logo di Google? Quali sono i colori principali utilizzati in HEX?



Il logo di Google utilizza principalmente il colore #4285F4 (blu) e #FBBC05 (giallo).



all rights reserved

# what do we need? people!

ethics, philosophy, morality... pragmatism

---

1. are well versed towards innovation
2. are not loaded with pre-existing clerical skepticism
3. can spot the difference between realistic and inflated contributions
4. are not scared of deep diving into the mechanics
5. have sensibility towards ethics
6. that believe in tools that enable professionals, rather than replace them

**WE NEED YOU!**





thanks ;)

✉ [michele.filannino@prometeia.com](mailto:michele.filannino@prometeia.com)

# Confidentiality

---

Any partial or total reproduction of its content is prohibited without written consent by Prometeia.

*Copyright © 2023 Prometeia*

# Contacts

---

## Bologna

P.zza Trento e Trieste, 3  
+39 051 6480911  
italy@prometeia.com

## London

Dashwood House 69 Old Broad Street  
EC2M 1QS  
+44 (0) 207 786 3525  
uk@prometeia.com

## Cairo

Smart Village - Concordia Building, B2111  
Km 28 Cairo Alex Desert Road  
6 of October City, Giza  
egypt@prometeia.com

## Milan

Via Brera, 18  
Viale Monza, 265  
+39 02 80505845  
italy@prometeia.com

## Istanbul

River Plaza, Kat 19  
Büyükdere Caddesi Bahar Sokak  
No. 13, 34394  
| Levent | Istanbul | Turkey  
+ 90 212 709 02 80 – 81 – 82  
turkey@prometeia.com

## Moscow

ul. Ilyinka, 4  
Capital Business Center Office 308  
+7 (916) 215 0692  
russia@prometeia.com

## Rome

Viale Regina Margherita, 279  
italy@prometeia.com



Prometeia



@PrometeiaGroup



Prometeiagroup



Prometeia

[www.prometeia.com](http://www.prometeia.com)



# OpenAI genesis

from the very beginning to today

- Dec. 2015 – OpenAI Inc. is **funded as a non-profit organization**, with US\$1B donations\* from Silicon Valley investors (Peter Thiel, Reid Hoffman, Sam Altman, Elon Musk, Jessica Livingston, Greg Brockman, Infosys, YC Research and Amazon Web Services (AWS)). The organization stated it would “freely collaborate” with other institutions and researchers by making its patents and research open to the public.
- Apr. 2016 – public beta of OpenAI Gym, a platform for reinforcement learning research
- Dec. 2016 – public release of Universe, a platform for measuring and training AGI across games, websites and apps
- Feb. 2018 – Elon Musk resigns from the board (remained donor) for conflict of interest with Tesla AI self-driving car division
- Jun. 2018 – Alec Radford & co. publish a pre-print about generative pre-training transformer (**GPT**) for a language model
- Feb. 2019 – **GPT-2** is announced, with only limited demonstrative versions released to the public. The complete version was not released out of concern over potential misuse, including applications for writing fake news.
- May 2019 – OpenAI **transitioned to “capped” for-profit** (100X on any investment). OpenAI LP is funded, receiving US\$1B investment from Microsoft and Matthew Brown Companies. OpenAI announces the intention to commercially license its tech
- Nov. 2019 – OpenAI releases the complete version of the GPT-2 language model
- Jun. 2020 – OpenAI announces **GPT-3**, a language model trained on trillions of words from the Internet, and “the API”, the heart of its first commercial product
- Sep. 2020 – GPT-3 is licensed exclusively to Microsoft.
- Jan. 2021 – OpenAI introduces **DALL-E**, a deep learning model that can generate digital images from natural language descriptions
- Sep. 2022 – OpenAI presents **Whisper**, a neural network that approaches human level robustness and accuracy on English speech recognition
- Dec. 2022 – launches a free preview of **ChatGPT**, a new AI chatbot based on GPT-3.5. It took just 5 days to reach 1M registered users (it took ~300 days to Facebook, ~150 days to Spotify, ~75 days to Instagram)
- Jan. 2023 – leaked documents show that Microsoft was in talks to invest US\$10B into OpenAI for 49% stake in, which **would be valued US\$29B**, and close the deal by 2022. Microsoft announces the integration of OpenAI services, ChatGPT included, in Azure later on.

# how it's been made?

from GPT-3 to ChatGPT (based on GPT3.5\*)

Step 1

Collect demonstration data,  
and train a supervised policy.

A prompt is  
sampled from our  
prompt dataset.

Explain the moon  
landing to a 6 year old

A labeler  
demonstrates the  
desired output  
behavior.

Some people went  
to the moon...

This data is used  
to fine-tune GPT-3  
with supervised  
learning.



Step 2

Collect comparison data,  
and train a reward model.

A prompt and  
several model  
outputs are  
sampled.

Explain the moon  
landing to a 6 year old

A > B > C > D  
Explain gravity... Explain not...  
Moon is natural satellite of... People went to the moon...

A labeler ranks  
the outputs from  
best to worst.

D > C > A > B

This data is used  
to train our  
reward model.



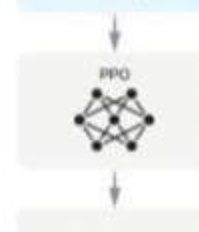
Step 3

Optimize a policy against  
the reward model using  
reinforcement learning.

A new prompt  
is sampled from  
the dataset.

Write a story  
about frogs

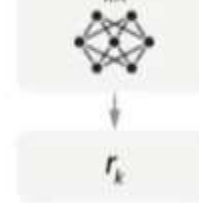
The policy  
generates  
an output.



The reward model  
calculates a  
reward for  
the output.

Once upon a time...

The reward is  
used to update  
the policy  
using PPO.



## REINFORCEMENT LEARNING FROM HUMAN FEEDBACK (RLHF)

Reinforcement Learning from Human Feedback is a subfield of Reinforcement Learning (RL) that **incorporates guidance from human experts** (feedback) in order to improve the learning and performance of an RL agent. This feedback can take many forms, such as providing explicit rewards or corrections, or simply demonstrating the correct behaviour. The goal is to improve the sample efficiency and overall performance of the RL agent, as well as to ensure that the learned behaviour aligns with human preferences and values.



# features & capabilities

for data engineers

- **Building boilerplate code**
- Add comments to your code
- Debug basic errors on your scripts
- Convert code from a programming language to another
- Process Flow Chart Creator



Create boiler plate code with python airflow for a process to read a file from redshift, add a new date column to the data and the save the file as a parquet to s3



Here is a sample Airflow DAG that demonstrates how you can use Python and the Airflow `RedshiftToS3Operator` to read a file from Redshift, add a new date column to the data, and save the file as a Parquet file to S3:

Copy code

```
import datetime

from airflow import DAG
from airflow.operators.redshift_to_s3_operator import RedshiftToS3Operator
from airflow.operators.python_operator import PythonOperator

default_args = {
    'owner': 'me',
    'start_date': datetime.datetime(2022, 1, 1)
}

dag = DAG(
    'my_dag_id',
    default_args=default_args,
    schedule_interval=datetime.timedelta(hours=1)
)

# This operator will read the data from Redshift and save it to a temporary
location on S3
read_from_redshift = RedshiftToS3Operator(
    task_id='read_from_redshift',
    redshift_conn_id='redshift_conn_id', # Replace with your Redshift
connection ID
    s3_conn_id='s3_conn_id', # Replace with your S3 connection ID
    sql='SELECT * FROM my_table', # Replace with your SQL query
    s3_bucket='my_bucket', # Replace with your S3 bucket name
    s3_key='tmp/data.csv', # Replace with your desired S3 key
    dag=dag
)
```

# features & capabilities

for data engineers

- Building boilerplate code
- **Add comments to your code**
- Debug basic errors on your scripts
- Convert code from a programming language to another
- Process Flow Chart Creator



Here is the code with added comments:



Copy code

```
import PyPDF2 as pypdf
import os

# Set the directory where the PDF files are located
directory = '../test-pdfs'

# Iterate through all the files in the directory
for file in os.listdir(directory):
    # Skip any files that are not PDFs
    if not file.endswith(".pdf"):
        continue
    # Open the PDF file
    with open(os.path.join(directory, file), 'rb') as pdfFile:
        # Create a PDF file reader object
        pdfReader = pypdf.PdfFileReader(pdfFile)
        # Get the number of pages in the PDF
        numpages = pdfReader.numPages
        # Iterate through all the pages
        for i in range(numpages):
            # Get the current page object
            pageObj = pdfReader.getPage(i)
            # Extract the text from the page
            a = pageObj.extractText()
            # Open a text file with the same name as the PDF file, in append
            mode
            file1 = open(f"{file}.txt", "a")
            # Write the extracted text to the text file
            file1.write(a)
            # Close the text file
            file1.close()
```

Regenerate response

# features & capabilities

for data engineers



- Building boilerplate code
- Add comments to your code
- **Debug basic errors on your scripts**
- Convert code from a programming language to another
- Process Flow Chart Creator




Debug my code

```
SELECT Customercity,  
    AVG(Orderamount) OVER(PARTITION BY Customercity) AS AvgOrderAmount,  
    MIN(OrderAmount) OVER(PARTITION BY Customercity) AS MinOrderAmount  
    SUM(Orderamount) OVER(PARTITION BY Customercity) TotalOrderAmount  
FROM [dbo].[Orders];
```



It looks like there is a syntax error in your code. The "SELECT" statement is missing a comma between the "MIN(OrderAmount)" column and the "SUM(Orderamount)" column. Here is the corrected version of your code:  

```
SELECT Customercity,  
    AVG(Orderamount) OVER(PARTITION BY Customercity) AS AvgOrderAmount,  
    MIN(OrderAmount) OVER(PARTITION BY Customercity) AS MinOrderAmount,  
    SUM(Orderamount) OVER(PARTITION BY Customercity) TotalOrderAmount  
FROM [dbo].[Orders];
```

 Copy code

This should fix the syntax error and allow you to run the query successfully.

It's always a good idea to carefully review your code for syntax errors before running it. You can use a text editor or an integrated development environment (IDE) with syntax highlighting to make it easier to spot errors.

# features & capabilities

for data engineers

- Building boilerplate code
- Add comments to your code
- Debug basic errors on your scripts
- **Convert code from a programming language to another**
- Process Flow Chart Creator



Here is the equivalent code in Scala:

Copy code

```
import sys.process._
import com.amazonaws.services.glue.{DynamicFrame, GlueContext}
import com.amazonaws.services.glue.util.{GlueArgParser, Job}
import org.apache.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]) {
    val sc: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(sc)
    val spark = glueContext.getSparkSession

    val args = GlueArgParser.getResolvedOptions(sysArgs,
Seq("JOB_NAME").toArray)
    val job = new Job(glueContext, args("JOB_NAME"))

    val inputDf = glueContext.getSourceWithFormat(
      connectionType = "s3",
      options = Map("paths" -> Seq("s3://test-bucket-glue/testing-csv")),
      format = "csv"
    ).getDynamicFrame()

    val outputDf = glueContext.getSinkWithFormat(
      connectionType = "s3",
      options = Map("path" -> "s3://test-bucket-glue/testing-output"),
      format = "parquet"
    ).writeDynamicFrame(inputDf)
  }
}
```



all rights reserved

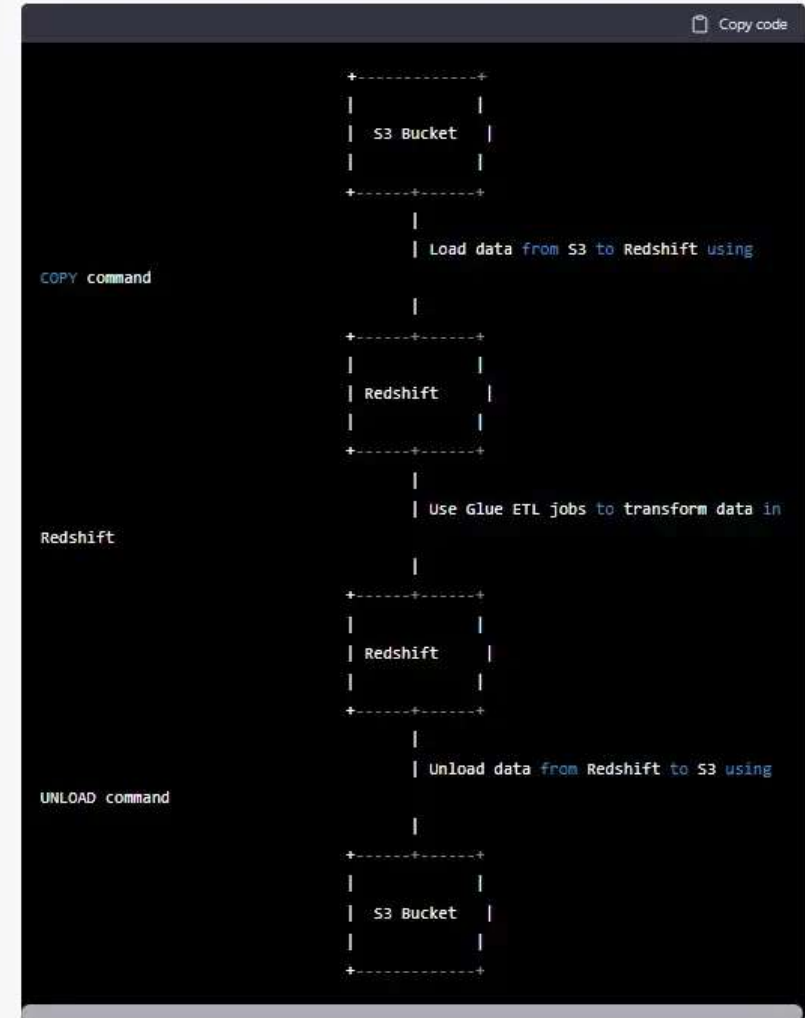
# features & capabilities

for data engineers

- Building boilerplate code
- Add comments to your code
- Debug basic errors on your scripts
- Convert code from a programming language to another
- **Process Flow Chart Creator**



Here is a flow chart of a process that loads data from S3 to Amazon Redshift, uses AWS Glue for transformation, and then unloads the data back to S3:  



I hope this helps! Let me know if you have any questions.