Extract PDF using Azure OpenAI

Created by David Song, May 26, 2025

Contents

[Use Model gpt-4o 1](#_Toc199269416)

[Source Code: 1](#_Toc199269417)

[Document Intelligence 2](#_Toc199269418)

[Create Document Intelligence resource 2](#_Toc199269419)

[Required Packages: 2](#_Toc199269420)

[Parsing Document 2](#_Toc199269421)

[Extract Table under a Header line 3](#_Toc199269422)

[Embedding 3](#_Toc199269423)

[Reference: 3](#_Toc199269424)

# Use Model gpt-4o

Required Packages:

langchain

langchain-community

PyPDFLoader

## Source Code:

from langchain\_openai import AzureChatOpenAI  
from langchain.prompts import PromptTemplate  
from langchain.chains import LLMChain  
from langchain\_community.document\_loaders import PyPDFLoader  
import json  
  
  
import json  
from dotenv import load\_dotenv  
import os  
load\_dotenv(r'C:\\*\*\*\Azure\.env')  
print(os.environ)

llm = AzureChatOpenAI(  
 azure\_deployment="gpt-4o", *# or your deployment* api\_version="2025-01-01-preview", *# or your api version*)  
  
  
loader=PyPDFLoader(r'C:\\*\*\*\Azure\2024-dfast-results-20240626.pdf')  
docs=loader.load()  
  
template="""  
you are an intelliegnt bot who can analyze any text with hospital information{doc\_text} ,   
your job is to read and analyse the information and create a json dictionary.  
dictionary has the followinmg key:  
**Table A.31. Wells Fargo & Company**  
output must be json nothing else  
"""  
prompt=PromptTemplate(template=template,input\_variables=["doc\_text"])  
llmchain=LLMChain(llm=llm,prompt=prompt)  
for doc in docs:  
 text=doc.page\_content  
 response=llmchain.invoke({"doc\_text":text})  
 data=response["text"]  
 data=data.replace("json","")  
 data=data.replace("`","")  
 *# data=json.loads(data) # check if data is emtpy* print(data)

# Document Intelligence

## Create Document Intelligence resource

## Required Packages:

pip install azure-ai-documentintelligence==1.0.0b4

## Parsing Document

mport os  
import base64  
from azure.core.credentials import AzureKeyCredential  
from azure.ai.documentintelligence import DocumentIntelligenceClient  
from azure.ai.documentintelligence.models import AnalyzeResult  
from azure.ai.documentintelligence.models import AnalyzeDocumentRequest  
  
AZURE\_DOCINT\_ENDPOINT = "https://\*\*\*.cognitiveservices.azure.com/"  
AZURE\_DOCINT\_KEY = "F1MGILsighFOHo\*\*\*"  
  
document\_intelligence\_client = DocumentIntelligenceClient(  
 endpoint=AZURE\_DOCINT\_ENDPOINT, credential=AzureKeyCredential(AZURE\_DOCINT\_KEY)  
 )  
  
'''  
# read a url file:  
poller = document\_intelligence\_client.begin\_analyze\_document("prebuilt-layout",   
 AnalyzeDocumentRequest(url\_source='https://raw.githubusercontent.com/sample-layout.pdf'))  
'''  
*#  
# read a local file  
#*input\_file = r"C:\work\Azure\sample-layout.pdf"  
with open(input\_file, "rb") as f:  
 base64\_encoded\_pdf = base64.b64encode(f.read()).decode("utf-8")  
  
analyze\_request = {  
 "base64Source": base64\_encoded\_pdf  
}  
poller = document\_intelligence\_client.begin\_analyze\_document("prebuilt-layout", analyze\_request=analyze\_request)  
result:AnalyzeResult = poller.result()

The result object is described in [2 AnalyzeResult Class Properties]

## Extract Table under a Header line

page\_number, header\_bounding\_polygon = find\_headerline\_region(result, header\_text)  
table = identify\_adjacent\_table(result, page\_number, header\_bounding\_polygon)  
df = extract\_table(table)

# Embedding

You will need to create a text-embedding-3-large model in order to use OpenAI embedding.

# Reference:

1. Get started with Document Intelligence: <https://docs.azure.cn/en-us/ai-services/document-intelligence/quickstarts/get-started-sdks-rest-api?view=doc-intel-4.0.0&pivots=programming-language-python>
2. AnalyzeResult Class Properties: <https://learn.microsoft.com/en-us/python/api/azure-ai-documentintelligence/azure.ai.documentintelligence.models.analyzeresult?view=azure-python>