Optimizing Everyday Tasks with CrewAI

Foad Kesheh · Following

7 min read · Jan 23, 2024

132 1

In the realm of AI frameworks, CrewAI stands out as a beacon of simplicity and power, designed primarily for engineers. At its core, CrewAI

orchestrates role-playing, autonomous AI agents, fostering collaborative intelligence to tackle complex tasks seamlessly. This framework, deeply rooted in the idea of collaborative intelligence, empowers agents to work in unison, making it an ideal tool for diverse applications.

The Mechanics of CrewAI

CrewAI operates on four fundamental building blocks: Agents, Tasks, Tools, and Crews. Each agent in CrewAI is like a dedicated team member with unique roles, backstories, goals, and memories. Tasks represent specific missions assigned to agents, while Tools are the means through which these agents accomplish their tasks. Finally, Crews are the collaborative spaces where Agents, Tasks, and Processes converge, leading to the execution of complex tasks.

CrewAI and LangChain: A Synergistic Relationship

tasks, from email management to complex research. The compatibility with LangChain significantly expands CrewAI's usability and functionality.

Exploring CrewAI: From Concept to Execution

Imagine a task like reading a PDF and creating a concise, well-structured article. With CrewAI, this process can be streamlined efficiently.

1. CrewAI installation is very simple, you only need a Python 3.10 environment and pip to install the required packages.

2. To create the CrewAI code, I implemented a GPT to streamline the process (https://chat.openai.com/g/g-AVGuUpRFb-crewai-code-generator)
3. We will pass it this prompt: I want to create a Crew that can reads a PDF URL and create an article very concise with 8–10 paragraphs, with a good title and a call to action to follow me on twitter. Ask me to input the PDF URL and my Twitter URL. If you like check here the result.
4. We will copy and paste the python code in a file called summarizer.py

```
# summarizer.py

from dotenv import load_dotenv
from crewai import Agent, Task, Crew, Process
from langchain_openai import ChatOpenAI
from langchain.tools import tool
import requests
from PyPDF2 import PdfReader
import re

# Load your OPENAI_API_KEY from your .env file
load_dotenv()

# Choose the model for the agents
model = ChatOpenAI(model_name="gpt-4-1106-preview", temperature=0.2)

# Tool to fetch and preprocess PDF content
@tool
def fetch_pdf_content(url: str) -> str:
    """
    Fetches and preprocesses content from a PDF given its URL.
    Returns the text of the PDF.
    """
    response = requests.get(url)
    with open('temp.pdf', 'wb') as f:
        f.write(response.content)

    with open('temp.pdf', 'rb') as f:
        pdf = PdfReader(f)
        text = '\n'.join(page.extract_text() for page in pdf.pages if page.extra

    # Optional preprocessing of text
```

```
    processed_text = re.sub(r'\s+', ' ', text).strip()
    return processed_text

# Agents
# PDF Reader Agent
pdf_reader = Agent(
    role='PDF Content Extractor',
    goal='Extract and preprocess text from a PDF',
    backstory='Specializes in handling and interpreting PDF documents',
    verbose=True,
    tools=[fetch_pdf_content],
    allow_delegation=False,
    llm=model
)

# Article Writer Agent
article_writer = Agent(
    role='Article Creator',
    goal='Write a concise and engaging article',
    backstory='Expert in creating informative and engaging articles',
    verbose=True,
    allow_delegation=False,
    llm=model
)

# Title Creator Agent
title_creator = Agent(
    role='Title Generator',
    goal='Generate a compelling title for the article',
    backstory='Skilled in crafting engaging and relevant titles',
    verbose=True,
    allow_delegation=False,
    llm=model
)

# CTA Integrator Agent
cta_integrator = Agent(
    role='Call to Action Integrator',
    goal='Incorporate a call to action to follow a Twitter account',
    backstory='Specializes in creating effective calls to action',
    verbose=True,
    allow_delegation=False,
    llm=model
)

# Tasks
def pdf_reading_task(pdf_url):
    return Task(
        description=f"Read and preprocess the text from the PDF at this URL: {pd
        agent=pdf_reader
```

```
    )

task_article_drafting = Task(
    description="Create a concise article with 8-10 paragraphs based on the extr
    agent=article_writer
)

task_title_generation = Task(
    description="Generate an engaging and relevant title for the article.",
    agent=title_creator
)

def task_cta_addition(twitter_url):
    return Task(
        description=f"Include a call to action at the end of the article, encour
        agent=cta_integrator
    )

# USER INPUTS
pdf_url = input("Enter the PDF URL: ")
twitter_url = input("Enter your Twitter URL: ")

# Instantiate and run the crew
crew = Crew(
    agents=[pdf_reader, article_writer, title_creator, cta_integrator],
    tasks=[pdf_reading_task(pdf_url), task_article_drafting, task_title_generati
    verbose=2
)

# Execute the crew
result = crew.kickoff()

# Combine results
final_article = f"{task_title_generation.output.result}\n\n{task_article_draftin
print("--------------------------")
print(final_article)
```

5. Create a .env file in the same folder and add your OpenAI API key there.

```
OPENAI_API_KEY=.....
```

6. Install the dependencies according to what the GPT provided

```
pip install python-dotenv crewai langchain-openai requests PyPDF2
```

7. Let's run our CrewAI, I will give it this article: https://arxiv.org/pdf/2401.03462.pdf and my Twitter

```
python summarizer.py
Enter the PDF URL: https://arxiv.org/pdf/2401.03462.pdf
Enter your Twitter URL: https://twitter.com/fkesheh84
```

Results

1. The first agent (pdf_reader) will grab the PDF content and preprocess it. This is the output of the first agent:

> *The extracted content from the PDF at the URL provided includes the abstract and sections of a paper titled "Soaring from 4K to 400K: Extending LLM's Context with Activation Beacon" by Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. The paper discusses the challenges of utilizing long contexts in large language models (LLMs) due to their limited context window length. The authors propose a method called Activation Beacon, which condenses the raw activations of LLMs into more compact forms, allowing the model to perceive longer contexts within a limited window. This method is introduced as a plug-and-play module for LLMs, preserving the model's original capabilities on short contexts while extending its ability to process longer contexts. The Activation Beacon is learned through an auto-regression task conditioned on a mixture of*

> *beacons with diversified condensing ratios. The paper includes detailed descriptions of the Activation Beacon's working mechanism, attention scheme, and learning method, as well as experimental studies demonstrating its effectiveness in extending context length and maintaining efficiency in both training and inference. The authors conclude that Activation Beacon presents an effective, efficient, compatible, and low-cost method for extending the context length of LLMs.*

2. The second agent (article_writer), creates the article body:

> *In the rapidly evolving field of artificial intelligence, large language models (LLMs) have made significant strides in understanding and generating human-like text. However, these models often hit a stumbling block when dealing with long contexts due to their limited context window length. In the groundbreaking paper "Soaring from 4K to 400K: Extending LLM's Context with Activation Beacon," researchers Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou tackle this issue head-on.*
>
> *The team introduces the Activation Beacon, a novel method that allows LLMs to condense the raw activations — essentially the model's internal responses — into more compact forms. This innovation enables the models to perceive and process much longer contexts than previously possible, all within the same limited window. The beauty of the Activation Beacon lies in its simplicity and versatility; it functions as a plug-and-play module that can be added to existing LLMs without compromising their performance on shorter texts.*
>
> *The Activation Beacon works through an auto-regression task, which is conditioned on a mixture of beacons with diversified condensing ratios. This means that the model learns to predict the next part of the text based on a*

condensed representation of the preceding text, allowing it to handle much larger chunks of information at once.

The researchers provide a detailed explanation of the Activation Beacon's inner workings, including its unique attention scheme and learning method. They delve into the technical aspects of how the beacon manages to maintain the delicate balance between condensing information and preserving the nuances necessary for accurate text generation.

To validate their approach, the authors conducted extensive experimental studies. The results are impressive: the Activation Beacon not only extends the context length that LLMs can handle but also maintains efficiency during both training and inference phases. This is a crucial development, as it ensures that the enhanced capabilities do not come at the cost of increased computational resources or slower processing times.

The paper concludes with the assertion that the Activation Beacon represents an effective, efficient, compatible, and low-cost solution to one of the most pressing challenges in the field of LLMs. By extending the context length, LLMs can now better understand and generate coherent and contextually rich text, opening up new possibilities for AI applications in areas such as conversation systems, document summarization, and beyond.

In summary, the Activation Beacon stands as a testament to the innovative spirit of AI research, offering a glimpse into a future where machines can understand and interact with human language in ways that were once thought impossible. As LLMs continue to grow in capability, tools like the Activation Beacon ensure that they remain both powerful and practical, ready to meet the complex demands of the real world.

3. The third agent (title_creator) will generate the title of our article

> *Breaking the Context Barrier: How the Activation Beacon Revolutionizes Large Language Models*

4. The last agent will create the call to action

> *Don't miss out on the forefront of technological innovation! Follow us at https://twitter.com/fkesheh84 for the latest insights and updates on how the Activation Beacon is revolutionizing large language models. Join our community of forward-thinkers and be part of the conversation. Click follow now and let's break barriers together!*

For sure this is a very simplified version of a crew, you might want to improve each agent descriptions and add it custom instructions tailored to your needs.

Conclusion: The Versatility of CrewAI

CrewAI, with its modular design and integration with LangChain tools, offers a robust platform for creating sophisticated AI agents capable of performing a wide range of tasks. Whether it's content creation, data analysis, or any other complex task, CrewAI's collaborative intelligence framework provides an efficient and effective solution.

For more in-depth information about CrewAI and its capabilities, you can explore their official website and GitHub repository.

Generative Ai Tools | Crew Ai | Ai Agent

Written by Foad Kesheh

Following

23 Followers

AI-First Full-Stack Developer | Exploring the Intersection of AI & Software Engineering | Sharing Insights & Latest Trends in AI | https://x.com/fkesheh84

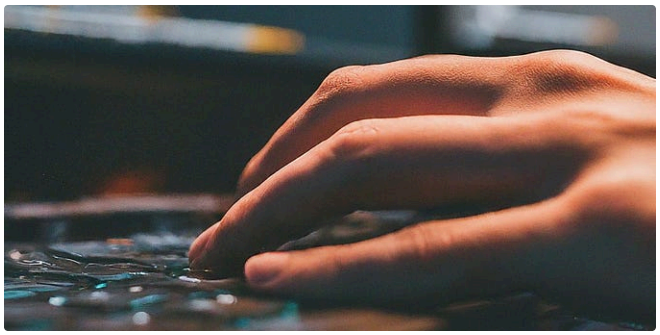
More from Foad Kesheh

Foad Kesheh

Enhancing Data Retrieval with Vector Databases and GPT-3.5...

This article explores integrating vector databases with GPT-3.5 for efficient data...

Feb 1 21

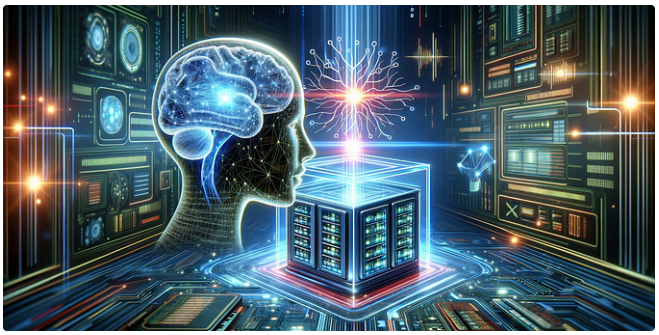


Foad Kesheh

Prompt Extraction Attack and Counter Measures for GPTs

Exploring the Vulnerabilities and Defenses of GPTs Against Prompt Extraction

Feb 8 52



Foad Kesheh

Streamlining Small Knowledge Bases: Harnessing LLMs for...

This article contrasts traditional RAG methods with an LLM-driven approach,...

Jan 26 50

Foad Kesheh

AI First Software Future

AI advances promise a future of intuitive, AI-crafted user interfaces and software,...

Feb 15

See all from Foad Kesheh

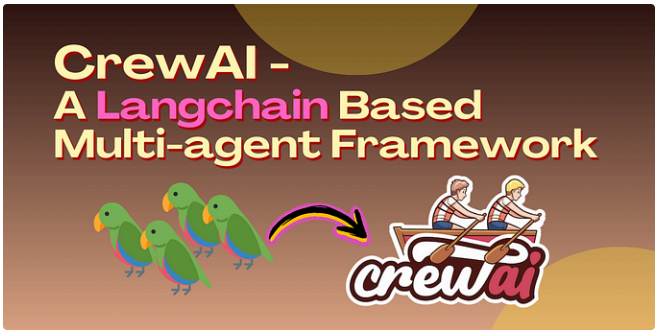
Recommended from Medium

Toon Beerten in Towards Data Science

Powerful Collaboration of AI Agents with CrewAI

A hands-on marketing use case

Feb 12 387 8



Yeyu Huang in Level Up Coding

For a Multi-Agent Framework, CrewAI has its Advantages...

A Quick Guide to the App Development with CrewAI and its comparison to AutoGen

Jan 28 578 2

Lists

Business

41 stories · 113 saves

Natural Language Processing

1476 stories · 987 saves

Medium's Huge List of Publications Accepting...

307 stories · 2782 saves

Staff Picks

649 stories · 996 saves

Csakash

Crew AI—you own minions

How I Made AI Assistants Do My Work For Me to Gather Tech News

Feb 26 64

Timur Taepov in Timur AI

I Tested AI Agents Team Using the CrewAI Framework

I finally got around to trying out the CrewAI framework hands-on, which is designed for...

Apr 22 11

Datadrifters in Generative AI

CrewAI in Action: Building and Orchestrating Your AI Dream Team

Everybody's talking about CrewAI, so let me clarify a few things.

Jan 22 596 5

Fatih KIR

How to create an AI team to write compelling stories with CrewAI a...

Are you fascinated by the idea of AI generating stories that capture the...

Feb 16 163 1

See more recommendations