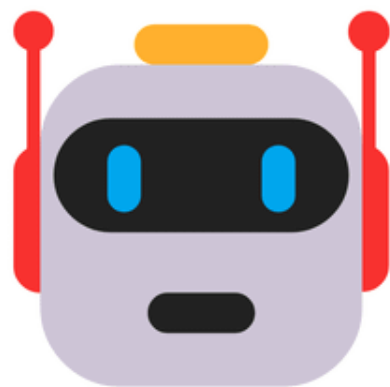


Researching and Writing blog posts with  
AI Agents! 🤖



+





Reads Paper



Writes Blog Post



Creates a Title



example.js

```
model = ChatOpenAI(  
    model="openhermes",  
    base_url="http://localhost:11434/v1"  
)
```



Agentic AI

```
# Tool for loading and reading a PDF locally
@tool
def fetch_pdf_content(pdf_path: str):
    """
    Reads a local PDF and returns the content
    """
    with open(pdf_path, 'rb') as f:
        pdf = PdfReader(f)
        text = '\n'.join(page.extract_text() for page in
pdf.pages if page.extract_text())

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





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



```
def pdf_reading_task(pdf):  
    return Task(  
        description=f"Read and preprocess the PDF at this local path: {pdf_local_relative_path}",  
        agent=pdf_reader,  
        expected_output="Extracted and preprocessed text from a PDF",  
    )  
  
task_article_drafting = Task(  
    description="Create a concise article with 8-10 paragraphs based on the extracted PDF content.",  
    agent=article_writer,  
    expected_output="8-10 paragraphs describing the key points of the PDF",  
)  
  
task_title_generation = Task(  
    description="Generate an engaging and relevant title for the article.",  
    agent=title_creator,  
    expected_output="A Title of About 5-7 Words"  
)
```





Agentic AI

```
# Instantiate and run the crew
crew = Crew(
    agents=[pdf_reader, article_writer, title_creator],
    tasks=[pdf_reading_task(pdf_local_relative_path),
           task_article_drafting,
           task_title_generation],
    verbose=2
)

# Let's start!
result = crew.kickoff()
```



# Transformers Unleashed: Harnessing Attention for Natural Language Processing

---

In "Attention Is All You Need" by Ashish Vaswani et al., a transformer architecture that utilizes an attention mechanism is proposed as the primary means for information interaction among its components. This attention mechanism allows the model to focus on relevant input elements and capture long-range dependencies in the input sequence, resulting in improved performance over