```
import os
from typing import List

from PyPDF2 import PdfReader

from swarms import Agent, OpenAIChat, SpreadSheetSwarm



def chunk_text(text: str, chunk_size: int = 400) -> List[str]:
    """
    Splits the input text into chunks of the specified size.

    Args:
        text (str): The text to be chunked.
        chunk_size (int): The size of each chunk.

    Returns:
        List[str]: A list of text chunks.
    """
    return [
        text[i : i + chunk_size]
        for i in range(0, len(text), chunk_size)
    ]



def select_central_chunks(
```

```
    chunks: List[str], keep_ratio: float = 0.7
) -> List[str]:
    """
    Selects the central portion of the chunks based on the keep_ratio.

    Args:
        chunks (List[str]): List of text chunks.
        keep_ratio (float): The ratio of chunks to keep, centered in the middle of the list.

    Returns:
        List[str]: A list of selected central chunks.
    """
    total_chunks = len(chunks)
    keep_count = int(total_chunks * keep_ratio)

    # Determine start and end index to keep the central portion
    start_index = (total_chunks - keep_count) // 2
    end_index = start_index + keep_count

    return chunks[start_index:end_index]


def convert_pdf_to_text(
    file_path: str, chunk_size: int = 300, keep_ratio: float = 0.5
) -> str:
    """
```

Converts a PDF into text, splits the text into chunks, selects the central chunks, and returns a single string.

```
    Args:
        file_path (str): The path to the PDF file.
        chunk_size (int): The size of each text chunk.
        keep_ratio (float): The ratio of the central chunks to keep.

    Returns:
        str: A string containing the selected central portion of the text.
    """
    with open(file_path, "rb") as pdf_file_obj:
        pdf_reader = PdfReader(pdf_file_obj)
        text = ""
        for page in pdf_reader.pages:
            text += page.extract_text()

    chunks = chunk_text(text, chunk_size)
    central_chunks = select_central_chunks(chunks, keep_ratio)

    # Concatenate the selected chunks into a single string
    return " ".join(central_chunks)


# Define custom system prompts for each agent
```

FINANCIAL_ANALYSIS_AGENT_SYS_PROMPT = """

You are an expert financial analyst specializing in interpreting corporate financial statements. Your task is to analyze the income statement, balance sheet, and cash flow statement from Nvidia's 10-K report. Identify key financial metrics, trends, and ratios that are critical for executive decision-making. Focus on revenue growth, profitability, liquidity, and solvency.

"""

RISK_ASSESSMENT_AGENT_SYS_PROMPT = """

You are an expert in risk management and financial compliance. Your task is to thoroughly review the risk factors outlined in Nvidia's 10-K report. Identify the most significant risks that could impact the company's financial performance, including market, operational, regulatory, and competitive risks. Provide an executive summary of the potential impact and mitigation strategies.

"""

MARKET_TRENDS_AGENT_SYS_PROMPT = """

You are a market analyst specializing in the semiconductor industry. Your task is to analyze market trends, competitive positioning, and external economic factors mentioned in Nvidia's 10-K report. Identify key opportunities and threats in the market, and provide insights on how these could influence Nvidia's strategic decisions.

"""

STRATEGIC_RECOMMENDATIONS_AGENT_SYS_PROMPT = """

You are a strategic consultant for executive leadership. Your task is to synthesize the analyses of financial performance, risks, and market trends from Nvidia's 10-K report. Based on these insights, provide actionable strategic recommendations for Nvidia's leadership team to enhance growth, manage risks, and capitalize on market opportunities.

"""

PDF_PARSING_AGENT_SYS_PROMPT = """
You are an expert in parsing and extracting data from complex documents. Your task is to extract the necessary financial statements, risk factors, and market analysis sections from Nvidia's 10-K report PDF. Provide the relevant text and data in a structured format for further analysis by other agents.
"""

REVENUE_MAXIMIZATION_AGENT_SYS_PROMPT = """
You are a revenue maximization specialist. Your task is to identify potential revenue growth opportunities for Nvidia based on the information provided in the 10-K report. Focus on innovative strategies, new markets, and product offerings that can drive revenue growth and enhance profitability.
"""

EXPENSES_MINIMIZATION_AGENT_SYS_PROMPT = """
You are an expert in minimizing expenses and optimizing cost structures. Your task is to analyze Nvidia's cost structure and operating expenses outlined in the 10-K report. Identify areas where cost savings can be achieved, operational efficiencies can be improved, and expenses can be minimized to enhance profitability.
"""

# Initialize the OpenAI model
api_key = os.getenv("OPENAI_API_KEY")
model = OpenAIChat(

```
    openai_api_key=api_key,
    model_name="gpt-4o-mini",
    temperature=0.0,
    max_tokens=4000,
)

# Initialize the agents for analyzing the 10-K report
agents = [
    Agent(
        agent_name="PDF-Parsing-Agent",
        system_prompt=PDF_PARSING_AGENT_SYS_PROMPT,
        llm=model,
        max_loops=1,
        saved_state_path="pdf_parsing_agent.json",
        user_name="swarms_corp",
        retry_attempts=1,
    ),
    Agent(
        agent_name="Financial-Analysis-Agent",
        system_prompt=FINANCIAL_ANALYSIS_AGENT_SYS_PROMPT,
        llm=model,
        max_loops=1,
        saved_state_path="financial_analysis_agent.json",
        user_name="swarms_corp",
        retry_attempts=1,
    ),
```

```
    Agent(
        agent_name="Risk-Assessment-Agent",
        system_prompt=RISK_ASSESSMENT_AGENT_SYS_PROMPT,
        llm=model,
        max_loops=1,
        saved_state_path="risk_assessment_agent.json",
        user_name="swarms_corp",
        retry_attempts=1,
    ),
    Agent(
        agent_name="Market-Trends-Agent",
        system_prompt=MARKET_TRENDS_AGENT_SYS_PROMPT,
        llm=model,
        max_loops=1,
        saved_state_path="market_trends_agent.json",
        user_name="swarms_corp",
        retry_attempts=1,
    ),
    Agent(
        agent_name="Strategic-Recommendations-Agent",
        system_prompt=STRATEGIC_RECOMMENDATIONS_AGENT_SYS_PROMPT,
        llm=model,
        max_loops=1,
        saved_state_path="strategic_recommendations_agent.json",
        user_name="swarms_corp",
        retry_attempts=1,
```

```
    ),
    Agent(
        agent_name="Revenue-Maximization-Agent",
        system_prompt=REVENUE_MAXIMIZATION_AGENT_SYS_PROMPT,
        llm=model,
        max_loops=1,
        saved_state_path="strategic_recommendations_agent.json",
        user_name="swarms_corp",
        retry_attempts=1,
    ),
    Agent(
        agent_name="Expenses-Minimization-Agent",
        system_prompt=EXPENSES_MINIMIZATION_AGENT_SYS_PROMPT,
        llm=model,
        max_loops=1,
        saved_state_path="strategic_recommendations_agent.json",
        user_name="swarms_corp",
        retry_attempts=1,
    ),
]

# Create a Swarm with the list of agents
swarm = SpreadSheetSwarm(
    name="Nvidia-10K-Analysis-Swarm",
      description="A swarm that analyzes Nvidia's 10-K report to provide executive-level financial insights and strategic recommendations.",
```

```
    agents=agents,
    autosave_on=True,
    run_all_agents=False,
    max_loops=1,
    workspace_dir="spreadsheet_workspace",
)

# Run the swarm to analyze the 10-K report
out = swarm.run(
    task=f"Analyze Nvidia's 10-K report and provide your analysis: {convert_pdf_to_text("10k.pdf", 200)}"
)
print(out)
```