

과제 결과 요약서

■ 과제 내용

이 과제는 AI 금융을 활용하여 환경 분야 투자에 도움을 줄 수 있는 인공지능 에이전트를 만드는 것입니다. 주요 기능으로는 환경 관련 뉴스 제공, 주식 정보 제공, 환경 관련 정보 제공, 환경 통계 관련 자료 제공이 있습니다.

이 챗봇은 Streamlit을 이용해 구현되었으며, LangChain의 Prompt template과 LLM을 사용하여 질문에 답변을 생성합니다. 각 기능은 독립적인 에이전트로 구현되어 있으며, Topic classifier가 사용자의 입력을 분석하여 적합한 에이전트로 전달합니다. 이 프로젝트는 환경 분야에 특화된 인공지능 솔루션을 제공하여 신뢰할 수 있는 정보를 기반으로 더 나은 투자 결정을 돕는 것을 목표로 합니다.

■ 과제 목표

본 과제의 목표는 환경 분야에 대한 투자 결정을 지원하는 인공지능 기반의 챗봇을 개발하는 것입니다. 이를 통해 사용자가 신뢰할 수 있는 정보를 바탕으로 환경 관련 투자 결정을 내릴 수 있도록 돕고, 환경 보호와 지속 가능한 발전에 기여하는 것입니다.

■ 과제 수행 방법

시간을 다음과 같이 나누어서 수행하였습니다.

3월: 챗봇의 기능 및 구조 설정, 구현할 기능 목록 정리

4월: 강의자료로 주어진 ticker_news_fund_streamlit.py 파일 참고 뉴스 검색 agent와 주식 정보 제공 agent 구현, Topic classifier 함수 제작

5월: 중간고사 이후 RAG(특정 데이터를 GPT에게 제공) 관련 agent 구현, embedding을 이용한 PDF 데이터 처리 및 SQL을 통한 데이터 검색 기능 추가

■ 과제 최종 결과물

최종 결과물은 환경 분야에 특화된 인공지능 챗봇입니다. 이 챗봇은 다음과 같은 기능을 제공합니다:

1. 사용자 입력을 분석하여 적합한 정보를 제공하는 Topic classifier
2. 환경 관련 최신 뉴스를 제공하는 get_news_agent
3. 특정 회사의 주식 가격 및 금융 지표를 제공하는 get_stockprice_agent
4. 환경 법 및 규제에 대한 정보를 제공하는 get_environment_economy_information_agent
5. 환경 통계 자료를 제공하는 get_statistical_info_agent
6. 다른 agent에서 처리할 수 없는 질문을 처리하는 normal_agent

■ 기대효과 및 활용분야

기대효과는 다음과 같은 3가지가 있습니다.

1. 사용자에게 신뢰할 수 있는 환경 관련 정보를 제공하여 더 나은 투자 결정을 돕습니다.
2. 환경 보호와 지속 가능한 발전에 기여하는 투자 유도를 촉진합니다.
3. 인공지능 기술을 활용하여 사용자 경험을 향상시키고, 효율적인 정보 검색 및 분석을 가능하게 합니다.

활용분야 분야로는 다음 3가지가 있습니다.

1. 개인 및 기관 투자자들이 환경 관련 투자 결정을 내릴 때 유용하게 사용할 수 있습니다.
2. 환경 관련 기업이나 단체에서 정보 제공 도구로 활용할 수 있습니다.

과제 결과 보고서

■ 과제 내용

과제의 내용은 AI 금융에서 배운 내용을 토대로 환경 분야에 투자를 할 때 관련된 도움을 줄 수 있는 인공지능 agent를 만드는 것입니다. 이 과제는 다음과 같은 기능을 포함합니다.

1. 환경 관련 뉴스 제공

- 사용자가 원하는 주제의 최신 뉴스를 검색하고 제공합니다.
- 뉴스의 제목, 날짜, URL을 포함하여 결과를 제공합니다.

2. 주식 정보 제공

- 특정 회사의 주식 가격 및 금융 데이터를 요청 시 제공합니다.
- 주식 가격 그래프와 주요 금융 지표를 시각적으로 제공합니다.

3. 환경 관련 정보 제공

- 환경 경제학에 대한 질문에 환경 경제학 책의 내용을 기본으로 한 답변을 합니다.
- 관련된 자료를 바탕으로 신뢰할 수 있는 정보를 제공합니다.

4. 환경 통계 자료 제공

- 온실 가스 관련 통계 자료와 관련된 질문에 대한 답변을 제공합니다.

이 챗봇은 Streamlit을 이용해 구현되었으며, LangChain의 Prompt template과 LLM을 사용하여 다양한 질문에 대한 답변을 생성합니다. 각 기능은 독립적인 agent로 구현되어 있으며, Topic classifier가 사용자의 입력을 분석하여 적합한 agent로 전달합니다.

이 프로젝트는 환경 분야에 특화된 인공지능 솔루션을 제공함으로써, 사용자가 신뢰할 수 있는 정보를 기반으로 더 나은 투자 결정을 내릴 수 있도록 돕는 것을 목표로 합니다.

■ 과제의 필요성

현재 생성형 AI가 주목받으면서 금융업계에 새로운 바람이 불고 있습니다. 실제로 골드만 삭스의 경우 증권가의 주니어들에 대한 채용을 크게 줄이겠다고 말하기도 하였습니다. 또한 AI가 증권사 리포트를 써 주는 상황 역시 나타났습니다. 이는 언젠가 금융권에 취업하게 되었을 때 AI를 이용하게 될 가능성이 커지게 되었음을 의미합니다. 그러므로 경제학과 학생들은 단순히 해당 현상을 지켜보고 절망만 하는 것이 아니라 AI를 어떻게 사용해야 할지 익히고 그것을 자신의 무기로 삼아야 합니다. 즉, 자신에게 필요한 정보를 주는 AI를 만들고 이를 다룰 수 있어야 합니다.

이 과제는 앞서 언급한 능력을 함양하기 위한 훌륭한 과정입니다. 먼저, 환경이라는 특정한 도메인을 선택해 인공지능 agent의 전문성을 높였습니다. 목적에 맞는 agent를 만들기 위해 접근하여 그냥 챗지피티를 사용하는 것이 아닌 전문성이 있는 agent를 만들어서 더 전문적으로 사용할 수 있도록 하였습니다. 그리고 이를 챗 지피티에서 사용하는 것이 아니라 직접 해당 agent의 출력, 입력, 지시사항등을 지정하여 더 정제된 형태의 답을 할 수 있도록 해 주었습니다. 또한 function을 주어 챗지피티가 주어진 정보만을 내뱉는 기능이 아니라 그래프를 그릴 수 있도록 하는 등 기능을 더 확장했습니다. 마지막으로 RAG를 사용하여 environment chatbot에게 환경경제에 대한 더 자세한 정보를 제공하고 SQL을 통한 데이터 검색을 지원해 agent가 전문성을 가진 지식을 이용해 답을 살수 있도록 하였습니다. 그 결과 이 environment chatbot을 통해서 news를 검색하고 환경 경제 자료를 검색하는 등의 환경 관련 투자를 위해 필요한 정보들을 일반 chatgpt를 사용하는 것보다 질 높은 자료들로 얻을 수 있게 되었습니다.

■ 과제 목표

제 처음 목표는 재생에너지, 폐기물 관리 및 재활용 같은 환경 관련 분야에 대한 투자를 도와주는 chatbot을 만드는 것이었습니다. 그래서 저는 제 챗봇에 다음과 같은 5가지 기능을 넣으려고 했습니다: 원하는 주제의 뉴스를 가져오는 기능, 주식 관련 정보를 제공하는 기능, 환경 법이나 환경 규제 등 도움이 되는 지식을 담은 자료를 기반으로 질문에 답하는 기능, 환경 관련 통계자료를 제공하는 기능, 마지막으로 지금까지의 기능을 총괄적으로 사용하여 투자에 대한 조언을 할 수 있는 기능입니다.

궁극적인 목표는 사용자가 이 챗봇을 통해 환경 관련 기업에 투자할 수 있는 충분한 정보를 얻어 투자까지 이어지도록 하는 것이었습니다.

■ 과제 수행 실적

먼저 저의 처음 목표에서 넣으려고 했던 5가지 기능 중 뉴스, 주식 관련 정보 제공, 환경 경제 자료 제공, 통계자료를 통한 데이터 가져오기의 4가지 목표에 대한 기능들은 달성 완료했습니다. 그러나 부족한 점과 달성하지 못한 점이 있습니다.

먼저 부족한 부분에 대해 언급하겠습니다. 환경 관련 챗봇을 만든다고 했으나, 특정 도메인에 특화되지 않았다는 점이 아쉽습니다. 뉴스나 주식 가격을 찾는 agent들은 너무나 광범위한 범위를 다루고 있으며, PDF나 통계에 대한 질문을 다루는 agent들도 특정 자료에 대한 것만 찾을 수 있습니다. 그 결과, 환경이라는 주제는 사라지고 그냥 질문에 답을 하는 기능을 하는 결과물이 나왔습니다. 특히 이는 전문성이 있는 agent를 만들려는 이 과목의 가장 큰 목표를 너무나 충족시키지 못했다는 점에서 치명적인 실수였습니다.

나름대로 왜 이렇게 되었나 분석해 본 결과 처음 범위를 잡을 때 너무 크고 방대한 범위를 잡았다는 것을 깨달았습니다. 제가 듣는 수업에서 다른 팀들은 저와 달리 법 중 군법이라는 더 세부적인 범위를 다루거나 뉴스를 통해 주식 가격을 예측한다라는 특정한 목표를 가지고 있었습니다. 그러나 저는 환경 관련 정보라는 너무나 광범위한 주제를 택했고 어떻게 도움을 줄지도 정확하게 정하지 않은 상태로 제작을 시작했기에 배운 내용들을 적용하였을 뿐 실질적인 도움이 되는 gpt를 만들지 못했습니다. 해당 프로젝트를 발표할 때 교수님께서 이걸 통해서 어떤 걸 할 것인지 어떻게 돈을 벌 수 있을지 고민하면서 만들었어야 했는데 저에게는 저런 질문이 없이 그냥 배운걸 써보고 싶다는 생각만 있고 프로젝트를 진지하게 진행해야겠다는 다짐이 부족하여 이런 일이 발생한 것으로 추측됩니다.

그리고 두 개의 agent를 동시에 작동시키다 보니 정보를 제공해주는 시간이 너무 오래걸린다는 단점도 있었습니다. 그러나 이것은 저의 코드 뿐만 아니라 chatgpt의 속도 문제일 가능성 역시 존재할 것으로 예상됩니다. 그러나 확실히 agent를 1개로 통합해서 속도를 올릴 수 있었어야 했었다는 후회는 남았습니다.

구현을 하고 싶었지만 못한 부분은 두가지가 있습니다.

첫 번째로, 이름은 chatbot이지만 대화를 하려면 사용자와 나눈 대화를 기억하고 있어야 하는데 이를 어떻게 해야 할지 몰라서 구현하지 못했습니다. 사실 이 부분을 수업 가장 마지막에 배워서 적용을 하지 못했는데도 추후 프로젝트를 다시 수정하면서 구현해보도록 하겠습니다.

두 번째로, 교수님께서 지적해주셨던 문제와 관련된 내용입니다. 지금은 agent 수가 적어 if문으로 agent를 처리하고 있지만, agent가 많아지면 if문으로 처리하는 것에 한계가 있을 거라고 말씀해주셨습니다. 그래서 Topic classifier를 바꿔볼까 생각해 보았지만, 적절한 방법을 찾지 못해 if문을 계속 사용하게 되었습니다. 이 역시 추후 고쳐볼 예정입니다. 교수님께서 과제에 대한 발표를 할 때 agent를 1개만 두지 말고 agent에게 여러 function을 주어서 topic classifier가 없어도 통합 agent가 알아서 함수를 불러오도록 만들 수 있게 하라고 말씀하셨기 때문입니다. 이를 여름방학 동안 랭체인에 대한 공부를 더 한 뒤 더해볼 예정입니다.

■ 과제 수행 방법

팀원은 조장인 저 혼자여서 혼자 해당 과제를 진행했습니다. 먼저, 3월에는 어떤 챗봇을 만들고 싶은지 그리고 그 챗봇에서 구현하고 싶은 기능에는 무엇이 있는지 정리하였습니다.

그리고 4월에 강의자료로 주어진 `ticker_news_fund_streamlit.py` 파일의 코드를 참고해 원하는 주제를 입력하면 뉴스를 가지고 오는 agent와 주식 관련 정보를 요구하면 주식 가격의 그래프와 금융 데이터를 제공하는 agent를 만들었습니다. 그리고 이 두 agent사용을 사용자가 직접 하는게 아니라 질문을 입력하면 저절로 알맞은 agent가 사용되도록 `Toic_classifier` 함수를 제작했습니다.

5월에는 중간고사 이후에 배운 내용인 RAG 즉, 추가적인 데이터를 gpt에게 제공하는 형태의 agent들을 제작했습니다. embedding을 이용해 pdf를 `vectorspace`에 넣어 chatgpt가 이해할 수 있는 형태로 제작한 후 `chroma db`에 넣어서 chatbot에서 불러오도록 했고 `sqlalchemy`를 이용해 사용자가 주어진 data에 대한 질문을 하면 query를 통해 질문에 답할 수 있도록 하는 agent를 만들었습니다. 이를 통해 비록 실용성은 떨어졌지만 수업에서 배웠던 여러 agent에 넣을 수 있는 기능들을 구현할 수 있었습니다.

■ 과제 결과물

먼저 전체적인 과제에 대한 설명을 한 뒤 각 agent들의 수행 결과를 사진으로 첨부하겠습니다.

제가 만든 chatbot의 전체 작동 과정에 대해 설명드리겠습니다. 먼저 Streamlit을 통해 파일을 실행시킵니다. 사용자가 질문을 입력하면 Topic classifier가 해당 입력이 어느 agent function으로 가야 하는지 판단하고 그에 맞는 키워드를 반환합니다. 예를 들어 뉴스 검색 요청이 들어오면 'news'라는 키워드를 반환하고, 특정 회사의 주식 가격이나 금융정보를 찾으라는 요청이 들어오면 'stock_price'를 반환합니다. 그런 다음 main 함수에서 반환된 키워드를 보고 적합한 agent가 입력에 대한 답변을 하도록 합니다. 그리고 그 결과를 Streamlit 화면에 보여줍니다.

Eco chatbot

Ask:

Enter

처음 실행할 때 보이는 화면입니다. 보이는 Ask 부분에 질문을 입력하면 알맞은 agent function을 불러와 답을 밑에 적어줍니다.

그럼 이제 각 agent와 그 결과물들을 보여드리겠습니다.

1.normal_agent

먼저 소개할 것은 Topic_classifier와 normal_agent입니다. 이 두 함수는 기능만 다를 뿐 거의 유사한 형태를 가지고 있어서 함께 소개하겠습니다. normal_agent는 가장 기본적인 대답을 해주는 agent입니다. 다른 agent function들은 특정 주제에 특화되어 있지만, 사용자가 이러한 주제에 맞지 않는 입력을 했을

때 normal_agent가 답변을 생성합니다. 예를 들어, 서강대학교의 위치를 묻는 등 eco chatbot의 기능에 맞지 않는 입력이 들어오면 normal_agent에서 답변을 생성합니다. 해당 agent는 사용자가 해당 chatbot에 들어오리라고 예측되지 않은 질문을 하였을 때 이를 처리해주는 역할을 합니다.

Eco chatbot

Ask:

where is sogang university?

Enter

Answering....please wait

Hi, I am the General Information Chat Bot. I am here to assist you with any other questions you may have.

Sogang University is located in Seoul, South Korea.

위 사진은 normal agent를 사용한 결과물입니다. Top classifier가 해당 질문을 평가하고 normal agent에게 들어가야 할 질문이라고 평가되면 해당 질문을 normal agent에게 주고 normal agent는 자신을 소개한 뒤 사용자에게 질문의 답을 제시합니다.

Topic classifier 함수와 normal_agent 함수는 다음과 같은 과정으로 만들어졌습니다. 먼저, LangChain의 Prompt template을 이용해 템플릿을 만들었습니다. 이 챗봇은 대화를 주고받기보다는 사용자의 질문에 대한 답을 주도록 만들었기에 chatprompt template이 아닌 prompt template을 사용했습니다. 그런 다음 모델을 지정해서 LLM을 지정하고, pipe 문자를 이용해 만든 prompt와 LLM, 그리고 답을 string 형식으로 만들어주는 stroutputparser를 연결해 체인을 만들었습니다. 마지막으로, 만들어진 체인에 invoke 함수를 통해 답을 얻어 화면에 출력했습니다. Topic classifier의 경우, 그 답을 가지고 main 함수에서 if문으로 어느 agent를 선택할지 결정했고, normal_agent의 경우 해당 답을 화면에 출력했습니다.

```
def normal_agent(input):
    normal_template = """
    You are an agent who answers questions that do not correspond to the function of all other agents in the environment chatbot.
    Never answer information you don't know.
    Here is a passage or question you need to find answer:
    {input}
    """

    topic_prompt = PromptTemplate(input_variables=['input'], template=normal_template)

    llm = ChatOpenAI(model="gpt-3.5-turbo-0125")
    llm_chain = topic_prompt | llm | StrOutputParser()

    response = llm_chain.invoke({"input": input})

    return response
```

2. get_stock_price agent

두 번째로 소개시켜드릴 agent는 get_stock_price agent입니다. get_stock_price는 회사의 주식 가격이나 금융 데이터를 요청하는 입력이 들어왔을 때 주식 가격의 그래프와 회사의 금융 지표들을 알려주는 기능을 합니다. 해당 agent의 경우 normal agent와 달리 chatgpt에게 function을 제공하여 이를 통해 yfinance data를 사용할 수 있도록 하였습니다. 해당 agent는 수업에서 제공해 주셨던 ticker_news_fund_streamlit.py의 함수를 참고해서 파이썬 yfinance모듈에서 데이터를 가져올 수 있도록 만들었습니다. 그 중 제가 신경써 나름대로 고친 점이 두가지 있습니다. 첫째, 두 번째 과제에서는 한 agent가 get_data를 통해 뉴스와 주식 데이터를 모두 받아 판단을 내렸다면, 여기에서는 뉴스와 주식 데이터를 출력하는 agent를 따로 만들었습니다. 둘째, 금융 정보들을 파일에 저장하고 이를 읽어오지 않고, 받은 결과들을 하나의 dataframe에 정리한 다음 해당 dataframe을 chatgpt에게 content 형식으로 줄 수 있도록 json 형식으로 변환하여 제공하였습니다. 이렇게 함으로써, 숫자로 이루어진 데이터들을 dataframe 형식으로 가공한 후 json으로 제공하면 LLM이 분석하기 더 수월할 것이라고 판단했습니다.

아까 보셨던 입력창에 중국의 태양광 전지 제조 기업인 JINCO Solar holding의 주식 데이터를 알려달라고 하면 다음과 같은 대답을 보여줍니다.

Eco chatbot

Ask:

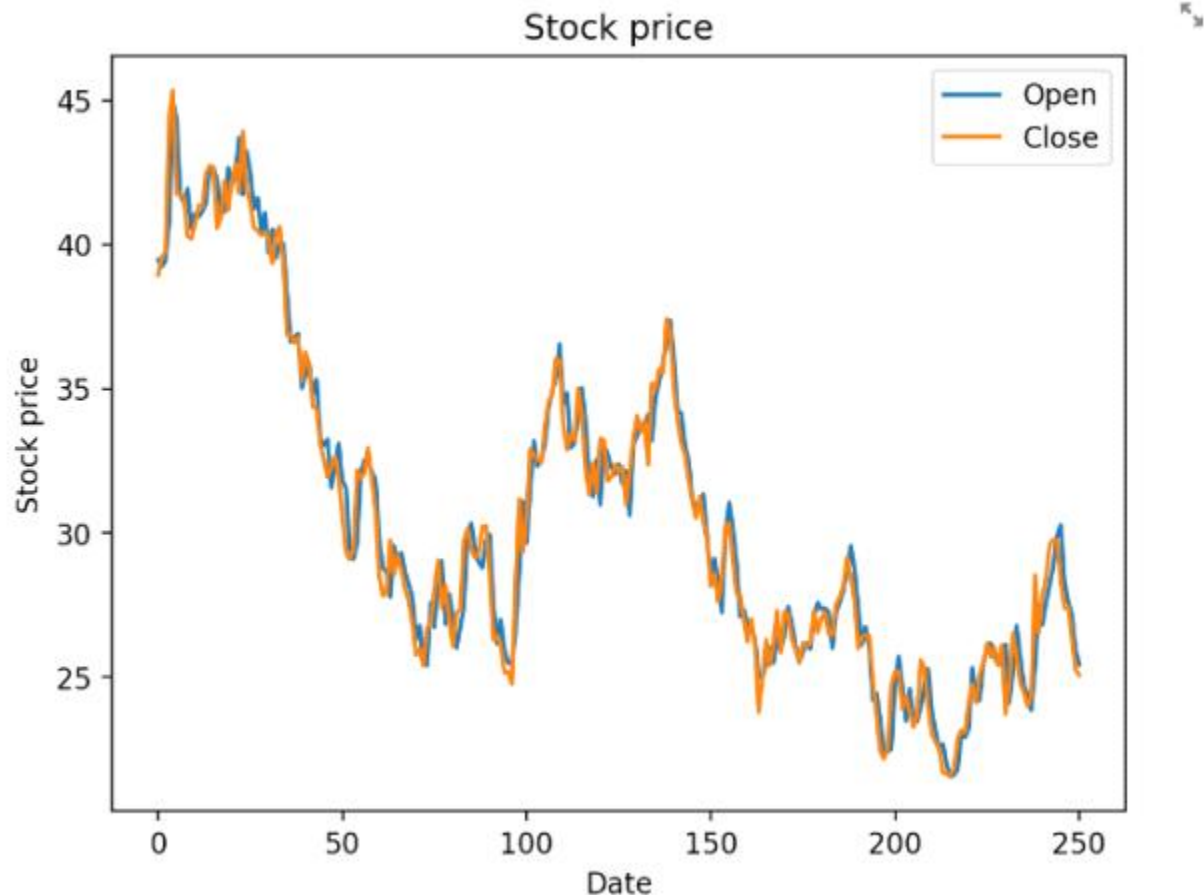
give me the stock data of jinko solar

Enter

Answering....please wait

Hi, I am the Stock Price Chat Bot. I will provide you with stock prices and financial data.

제일 먼저 자신이 stock price agent라는 것을 알립니다.



그 다음 yfinance에서 가져온 데이터를 이용해 주식 가격의 histogram을 그립니다. 그리고 얻어온 데이터들을 바탕으로 해당 회사의 주식 가격, 현금 흐름, 가치 평가, 재무상태표에 대한 정보를 제공합니다.

Stock Price

JinkoSolar Holding Co., Ltd. (JKS) is currently trading at 25.07. The stock has a 52-week range between 21.06 and 47.85. The previous close was 25.26, and today's high and low have been 25.55 and 24.85 respectively. It has a market cap of approximately 1.28 billion. The stock has a trailing P/E ratio of 3.47 and a forward P/E ratio of around 4.08. The average volume is 919,650 shares. The 50-day moving average stands at 25.10, while the 200-day moving average is \$28.77.

Cash Flow

As of the last fiscal year, JinkoSolar had free cash flow of -2.02 billion. The company paid interest of 1.11 billion and taxes of 1.78 billion. For the year ending December 31, 2023, the company's operating cash flow was 13.83 billion. Investing activities saw a cash outflow of -15.16 billion, and financing activities had a cash inflow of 8.64 billion. The overall change in cash resulted in an ending cash position of \$19.43 billion.

Valuation Measures ⇄

JinkoSolar has a beta of 0.42, indicating lower volatility compared to the market. The company has a trailing dividend rate of 10.636, equivalent to a dividend yield of around 4.2%. The enterprise value of the company stands at about 55 billion, with an enterprise value – to – revenue ratio of 0.465 and an enterprise value – to – EBITDA ratio of 3.785. The profit margins are at 2.7617.63 billion and total debt of \$53.47 billion.

Balance Sheet

On the balance sheet side, JinkoSolar's total assets amount to 135.83 billion, with total liabilities (net minority interest) being 102.3 billion. The company maintains a working capital of 1.88 billion with a total debt of 56.79 billion. Shareholder equity stands at 20.16 billion. The company's net tangible assets are valued at 17.76 billion. The company has a quick ratio of 0.527 and a current ratio of 1.025, indicating its short-term liquidity position.

Company Summary

JinkoSolar Holding Co., Ltd., based in Shangrao, China, engages in the design, development, production, and marketing of photovoltaic products. It's a leading player in the solar industry with an integrated annual capacity of 85 GW for mono wafers, 90 GW for solar cells, and 110 GW for solar modules. The company operates globally in regions like the USA, Mexico, Australia, Japan, and several European and Asian countries. Founded in 2006, JinkoSolar employs around 57,397 people and is a key player in solar power generation and energy storage.

3. get_news_agent

세 번째로 소개시켜드릴 agent는 get_news_agent입니다. get_news_agent는 특정한 주제의 뉴스를 요청하는 입력을 받을 때 뉴스를 가져오고, 다시 가져온 뉴스를 topic별로 나누어 뉴스의 제목과 날짜, 그리고 URL을 제공하는 agent입니다. get_news_agent도 강의 자료로 제공해주신

ticker_news_fund_streamlit.py의 get_news function을 참고했습니다. serf api key를 이용해 해당 주제의 뉴스를 받고 이를 제목, 날짜, 기사 URL, 기사 내용 요약의 형식으로 출력하게 했습니다. 또한 주제별로 뉴스를 모아서 제공할 것을 요구하여 사용자가 더 원하는 정보를 쉽게 얻을 수 있도록 설계하였습니다.

다음은 eco chatbot에게 환경 news를 요구하였을 때의 결과입니다. 해당 사진들을 보시면 알 수 있습니다. 싶이 뉴스를 주제별로 나누어 제공하고 있습니다.

Eco chatbot

Ask:

news about EU environment regulation

Enter

Answering....please wait

Hi, I am the News Chat Bot. I will provide you with the latest news.

Topic: EU Environmental Policy Threats and Reactions

Title: Europe's Swing to the Right Threatens Global Climate Policy

- **Date:** 2024-06-08
- **URL:** [Inside Climate News](#)
- **Summary:** The article discusses how recent rightward shifts in European politics could affect global climate policies, emphasizing potential rollbacks on existing environmental regulations.

Title: European farmers angry at climate policies could help sway EU parliamentary elections

- **Date:** 2024-06-09
- **URL:** [NPR](#)
- **Summary:** European farmers are expressing frustration over stringent climate policies, and their discontent could influence outcomes in upcoming EU parliamentary elections.

Title: EU Parliament approves final set of environmental regulations amid far-right threats to revoke green laws

- **Date:** 2024-06-10
 - **URL:** [EL PAÍS English](#)
 - **Summary:** The EU Parliament has approved a comprehensive suite of environmental regulations, despite threats from far-right factions to overturn these green laws.
-

Topic: Implementation of Specific EU Environmental Regulations

Title: New Regulation on waste shipments enters into force

- **Date:** 2024-06-08
- **URL:** [European Union](#)
- **Summary:** New regulations on waste shipments across the EU have come into effect, aiming to streamline waste management and reduce illegal waste trafficking.

Title: Gard: What's new with the revised EU Environmental Crime Directive

- **Date:** 2024-06-09
- **URL:** [safety4sea](#)
- **Summary:** The revised EU Environmental Crime Directive introduces new measures to combat environmental crimes, including stricter penalties and enhanced enforcement mechanisms.

Title: EU's new environmental laws irk developing countries

- **Date:** 2024-05-27
 - **URL:** [Financial Times](#)
 - **Summary:** Developing countries are critical of the EU's new environmental laws, arguing that these regulations impose unfair burdens on them and could hinder their economic growth.
-

Topic: General Overviews and Best Practices

Title: The European Union leads in climate change regulations

- **Date:** 2024-05-24
- **URL:** [Wolters Kluwer](#)
- **Summary:** The article highlights the EU's leadership in climate change regulation, detailing various policies and their impact on reducing carbon emissions.

Title: EU sustainability regulation

- **Date:** 2024-05-24
- **URL:** [Deloitte](#)
- **Summary:** Deloitte provides an overview of the latest EU sustainability regulations and their implications for businesses and industries across Europe.

Title: Air pollution | European Environment Agency's home page

- **Date:** 2024-06-09
- **URL:** [European Environment Agency](#)
- **Summary:** The European Environment Agency presents an in-depth analysis of air pollution issues, current regulations, and ongoing efforts to improve air quality in the EU.

Title: Best practices for printers to navigate new EU environmental regulations

- **Date:** 2024-06-03
- **URL:** [WAN-IFRA](#)
- **Summary:** This article offers practical advice for printing businesses to comply with new EU environmental regulations, focusing on sustainability and regulatory changes.

4. get_environment_economy_agent

다음으로 소개드릴 agent는 get_environment_economy_agent입니다.

get_environment_economy_information_agent는 환경 경제와 관련된 질문을 했을 때 질문의 답을 벡터 스페이스에서 찾고 이를 정제하여 답변을 주는 역할을 합니다. 이를 위해 환경 경제학과 관련된 지식을 embedding한 vectorspace가 필요하여 chatgpt_embedding.py라는 파이썬 코드를 사용해 embedding을 해주었습니다.

먼저, 환경 경제학 책을 EBSCO에서 찾아서 PDF로 다운로드 받았습니다. 알프레드 엔드리스의 "Environmental Economics: Theory and Policy"라는 책을 사용했습니다. 그 책을 PyPDFLoader로 불러오

고 문서를 split_document로 나눴습니다. 그리고 PDF의 내용들만 모아서 OpenAI에게 embedding을 요청해 embedding한 결과를 가져왔습니다. 그런 다음 해당 PDF의 텍스트와 페이지 수 같은 metadata들, 그리고 embedding 결과 등을 ChromaDB에 새로운 저장공간을 만들어서 저장해주었습니다. 이후 environment_chatbot 파일에서 agent를 사용할 때 ChromaDB에서 chatgpt_embedding.py에서 저장해둔 embedding을 가져오게 하여 이를 이용해 원하는 내용을 찾을 수 있도록 만들었습니다.

embedding을 사용한 function의 모습은 다음과 같습니다. 해당 벡터 스페이스를 직접 확인하기는 어려워 코드 사진을 첨부합니다.

```
loader = PyPDFLoader("./environmental_economics_100p.pdf")
docs = loader.load()
splitter = RecursiveCharacterTextSplitter(chunk_size=1000)
split_documents = splitter.split_documents(docs)

texts = [doc.page_content for doc in split_documents]

metadatas = [{"source": doc.metadata.get("source", ""), "page": doc.metadata.get("page", 0)} for doc in split_documents]

embeddings = get_openai_embeddings(texts)

client = chromadb.PersistentClient(path="/chroma_db")

collection = client.create_collection(name="environment")

collection.add(
    documents=texts,
    embeddings=embeddings,
    metadatas=metadatas,
    ids=[str(i) for i in range(len(texts))]
)
```

이를 통해 embedding한 책의 내용을 agent에 물어보면 해당 책의 내용에서 답을 찾아 사용자에게 전달해줍니다.

Answering....please wait

Hi, I am the Environmental Economics Chat Bot. I will provide you with information related to environmental economics.

The key difference between market equilibrium and social optimum in the context of environmental policy relates to the presence of externalities:

1. **Market Equilibrium:**

- **Definition:** The point at which the supply and demand curves intersect, resulting in an optimal allocation of resources based on individual preferences and production costs.
- **Consideration of Externalities:** Market equilibrium does not account for externalities, which are costs or benefits that affect third parties and are not reflected in market prices.

2. **Social Optimum:**

- **Definition:** The allocation of resources that maximizes social welfare, considering the total benefits to society, including externalities.
- **Consideration of Externalities:** Social optimum takes into account externalities, aiming to internalize them so that prices reflect the true social cost or benefit of goods and services.

When externalities are present, the market equilibrium (x^*) and the social optimum (x^{**}) differ. This divergence is known as a "market failure" in economics, where the market fails to allocate resources efficiently from a societal perspective.

Source: Environmental Economics Text, Pages 12 and 17

전문적인 환경 경제학의 책 내용을 포함하고 있기에 더 깊은 대답이 가능했습니다.

5. **get_statistical_info agent**

마지막으로 get_statistical_info agent에 대해 소개하고 그 결과물을 보여드리겠습니다.

get_statistical_info agent는 미리 넣어둔 온실가스 감지량 데이터에 대한 질문을 했을 때 작동합니다. 해당 자료는 기상청에서 제공하는 온실가스 감지량 월별 데이터의 2018년부터 2024년 데이터를 사용했습니다. 해당 데이터를 CSV 파일로 using_sqlalchemy.py라는 파일에서 SQLAlchemy를 이용해 저장하였습니다. using_sqlalchemy.py 함수의 실행과정은 다음과 같습니다. 먼저 파일을 읽어온 다음 경로를 지정해 해당 데이터를 저장할 폴더를 만들어 준 후, to_sql을 통해 엔진에 데이터를 저장해주었습니다. 이를 통해 해당 엔진에 쿼리를 넣어서 질문을 하면 해당 데이터에서 쿼리를 통해 데이터를 찾아올 수 있게 되었습니다.


```

import os
import pandas as pd
from sqlalchemy import create_engine

df = pd.read_csv("C:/Users/fpqhs/chatgpto/CLM_온실가스_2018-2024.csv", encoding='euc-kr')

db_path = "C:/Users/fpqhs/chatgpto/environment.db"

os.makedirs(os.path.dirname(db_path), exist_ok=True)

engine = create_engine(f"sqlite:///({db_path})")

df.to_sql("CO2_observation", engine, index=True)

```



environment db가 제가 지정한 폴더에 생성된 모습입니다. 이렇게 environment_chatbot에서 using_sqlalchemy 파일에서 저장한 데이터베이스, 즉 온실가스 통계량에 대한 표를 가져왔습니다. 그런 다음 LangChain의 create_sql_agent 함수를 통해 가져온 데이터베이스를 사용해 pandas ai처럼 문자열을 입력으로 넣었을 때 SQL로 변환하여 검색해주는 agent를 만들었습니다. 그리고 이 agent에 사용자의

입력을 넣은 다음 AI의 답을 반환해 화면에 띄워주었습니다. 다음은 2018년도부터 2024년도까지 독도와 울릉도에서 관측된 이산화 탄소의 최대양을 물어보는 문제에 대한 답입니다.

Ask:

Which month in 2018 recorded the highest average CO2 concentration at the Ulleungdo and Dokdo site?

Enter

Answering....please wait

Hi, I am the Statiscal Information Chat Bot. I know montly Co2 emission data from 2018 to 2024

The month in 2018 that recorded the highest average CO2 concentration at the Ulleungdo and Dokdo site was April, with an average CO2 concentration of 419.41 ppm.

이렇게 질문을 하면 맞는 agent함수를 호출해 답을 내놓는 environment_chatbot을 만들었습니다.

■ 기타

앞서 언급했다시피 해당 프로젝트는 보완할 것이 많은 프로젝트이기에 여름방학때 langchain과 도메인 지식에 대한 공부를 더 진행 한 뒤 보완할 예정입니다.