

# LangGraph + Agent Chat-bot

발표 일시 : 2025.05.20(화)

발표자 : 강승현

목차

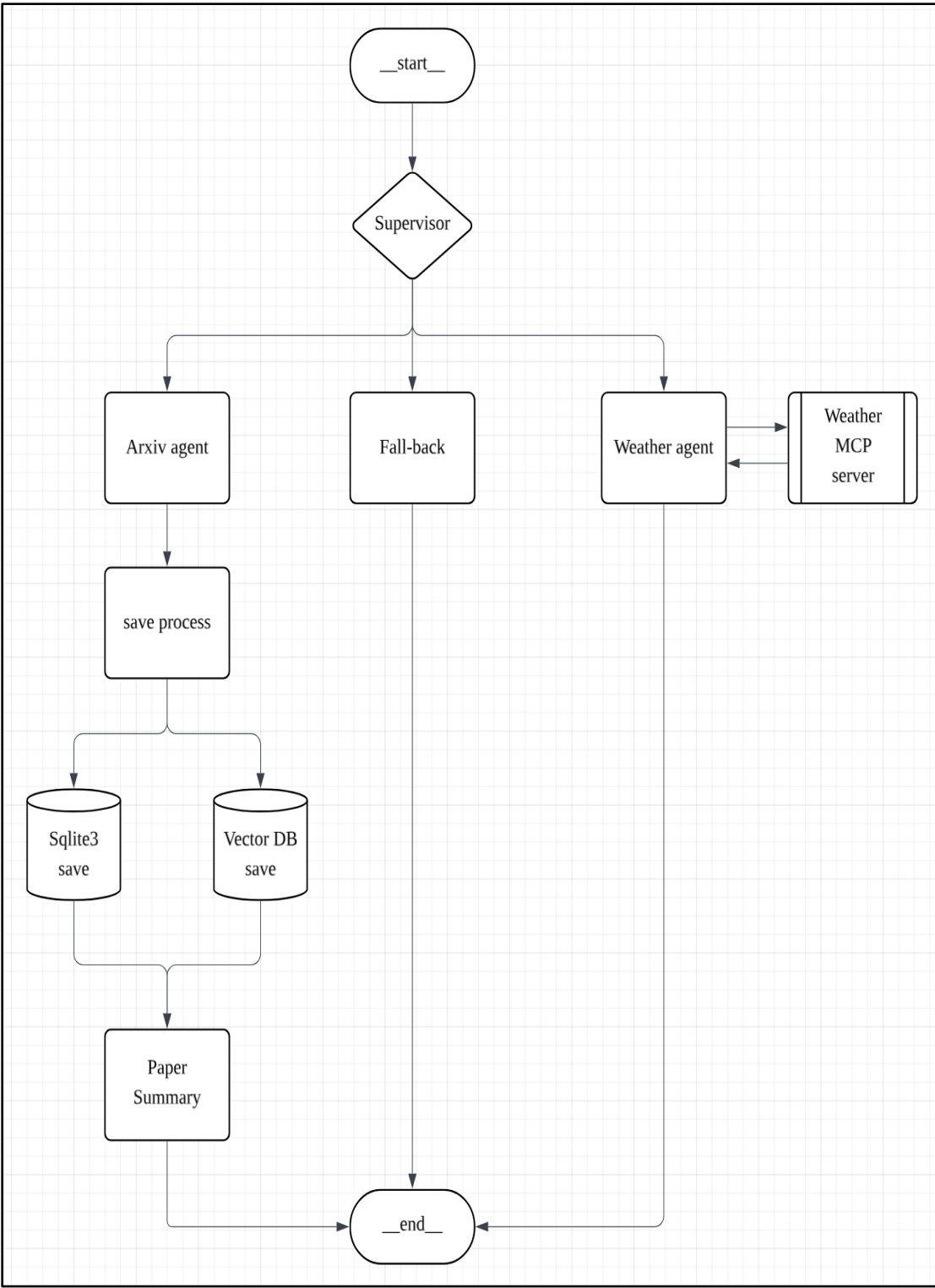
- 1. 프로젝트 개요
- 2. Graph 구조
- 3. 기능 소개
- 4. 한계
- 5. 추후 개발 사항
- 6. 후기

1. 프로젝트 개요

- LangChain 생태계를 활용하여 개인화된 Ai 비서를 만드는 것을 목표로 해당 프로젝트를 진행하였음.
- 해당 프로젝트는 LangGraph 기반 대화형 chat-bot이며 Agent 및 MCP에 대한 내용이 포함되었음.
- 대표적인 기능으로는 Arxiv agent와 Weather agent 두가지 종류가 있음.
  - Arxiv agent: Arxiv url을 입력 받아서 해당 논문 파일을 저장 및 요약해줌.
  - Weather agent: weather mcp server와 통신하여 사용자가 입력한 지역의 날씨를 알려줌.

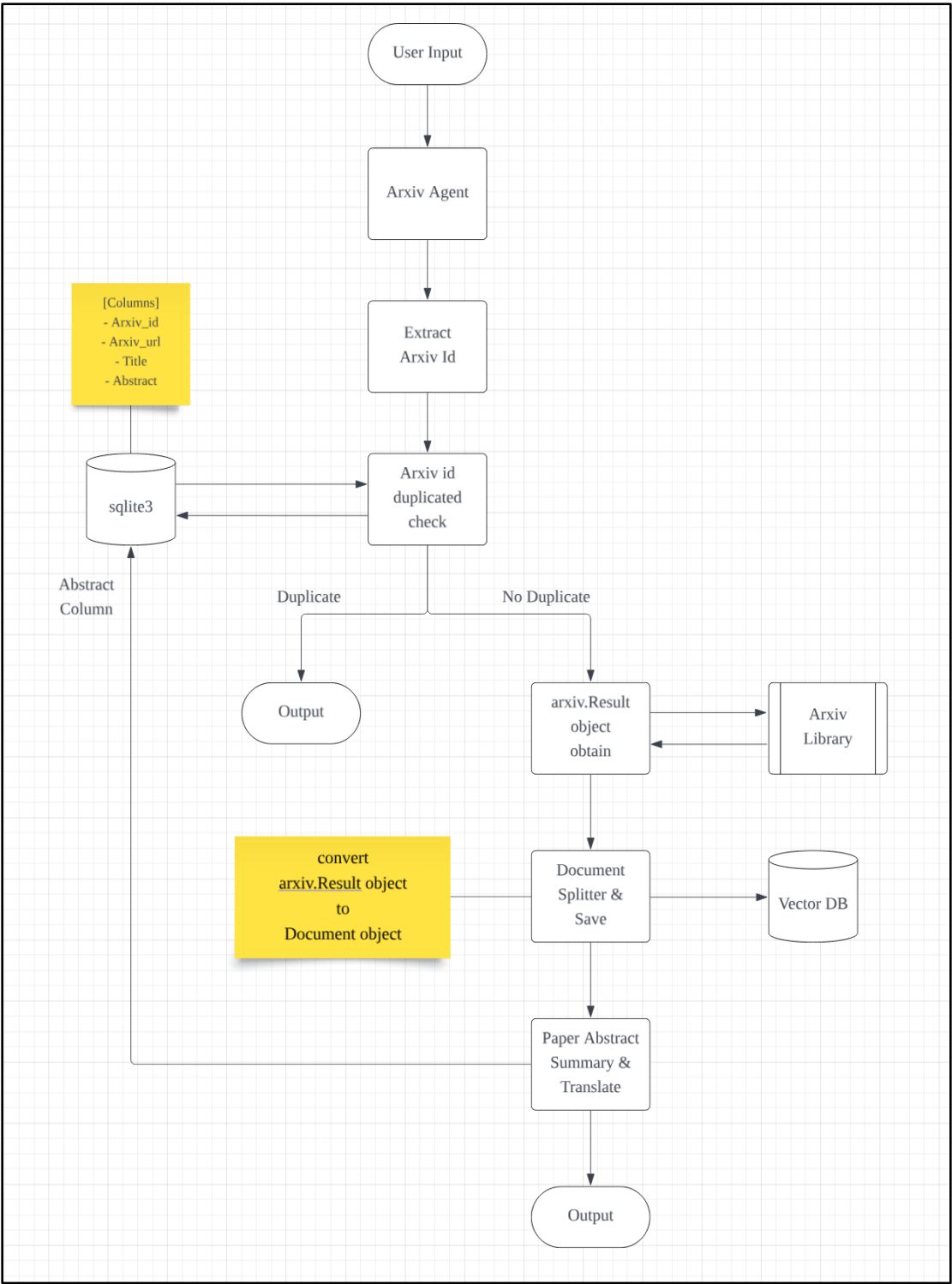
※ MCP(Model Context Protocol)이란?  
Anthropic에서 공개한 LLM과 외부 도구/서비스 사이의 표준화된 상호작용 프로토콜을 정의하는 오픈소스 프로토콜  
(Chat-gpt 또한 MCP를 제공하지만 api를 통해서만 사용이 가능.)

2. Graph 구조



- \_\_start\_\_: node가 가장 처음 시작되는 node.  
해당 node에서 user\_input을 memory로 저장함.
- Supervisor: route node. 다음 node로 보낼 곳을 정함.
- Arxiv agent: arxiv 관련 url을 입력 받는다면 해당 agent가 작업을 시작함.
- Weather agent: 날씨 관련된 입력을 받는다면 MCP server와 통신하며 날씨 정보를 출력함.
- Fall-back: arxiv와 날씨 외 모든 답변을 담당함.
- \_\_end\_\_: node가 종료되는 부분.  
해당 node에서 Ai message가 memory로 저장됨.

2. Graph 구조



- Arxiv Agent의 Flow chart
- Arxiv 라이브러리를 통해 논문 내용이 담긴 객체를 전달 받고 이를 Document 객체로 변환하여 Vector DB 및 Sqlite3 db에 저장함.

3. 기능소개

● Chat Memory

Lang-Secretary



안녕 우리집 강아지 이름은 철수야



안녕! 철수라는 이름 정말 귀엽다. 앞으로 철수에 대해 궁금한 점이나 이야기하고 싶은 게 있으면 언제든 말해줘!



우리집 강아지 이름이 뭐라고?



네가 말해준 우리집 강아지 이름은 "철수"야!

3. 기능소개

● Weather Agent

Lang-Secretary



오늘 강남구 날씨 알려줘





검색한 도시: 대한민국, 서울 강남구  
기준 시간: 19:53  
온도: 22.02도 (체감온도: 22.52도)  
날씨: 온흐림  
풍속: 3.36m/s

오늘은 온흐림에 22도 정도로 비교적 선선한 날씨입니다. 가벼운 긴팔이나 얇은 자켓, 혹은 반팔에 얇은 겹옷을 챙기시면 좋겠습니다. 바람이 조금 있으니, 야외 활동 시에는 가벼운 바람막이도 추천드려요!

3. 기능소개

● Weather Agent

 오늘 도쿄 날씨 알려줘



검색한 도시: 일본, 도쿄  
기준 시간: 20:10  
온도: 24.84도 (체감온도: 25.02도)  
날씨: 맑음  
풍속: 2.57m/s  
  
복장 추천: 오늘 도쿄는 밤에도 비교적 따뜻하고 맑은 날씨입니다. 가벼운 반팔이나 얇은 긴팔, 얇은 바지 또는 치마가 적당합니다. 바람이 약간 있으니 얇은 가디건이나 점퍼를 챙기면 더 좋겠어요.




3. 기능소개

● Arxiv Agent

Lang-Secretary



 <https://arxiv.org/abs/1706.03762> 다운 받아



Attention Is All You Need

Abstract

지배적인 sequence transduction 모델들은 encoder-decoder 구성에서 복잡한 recurrent 또는 convolutional neural network에 기반하고 있습니다.  
가장 성능이 뛰어난 모델들은 attention mechanism을 통해 encoder와 decoder를 연결하기도 합니다.  
우리는 recurrence와 convolution을 완전히 배제하고 attention mechanism만을 기반으로 하는 새로운 간단한 네트워크 아키텍처인 Transformer를 제안합니다.  
두 가지 기계 번역 과제에 대한 실험에서 이 모델들은 더 높은 품질을 보이면서도 병렬화가 더 용이하고 학습에 필요한 시간도 크게 줄어듦을 보여줍니다.

TimeLine

- 1. 1997년 - [Long Short-Term Memory \(LSTM\)](#)  
→ RNN의 장기 의존성 문제를 해결하며 sequence modeling의 대표적 구조로 자리잡음. Transformer는 이러한 recurrence 구조를 대체함.
- 2. 2014년 - [Sequence to Sequence Learning with Neural Networks](#)  
→ Encoder-Decoder 구조를 도입하여 기계번역 등 다양한 sequence transduction 문제에 적용. Transformer는 이 구조를 attention 기반으로 재해석함.
- 3. 2015년 - [Neural Machine Translation by Jointly Learning to Align and Translate \(Bahdanau Attention\)](#)  
→ Attention mechanism을 도입하여 encoder와 decoder 간의 정보 흐름을 개선. Transformer는 이 attention을 모델의 핵심으로 확장함.
- 4. 2017년 - [Attention Is All You Need](#)  
→ Recurrence와 convolution 없이 오직 attention만으로 sequence modeling을 수행하는 Transformer 아키텍처를 제안. 이후 NLP 및 다양한 분야에서 표준이 됨.



reference

- [Sequence to Sequence Learning with Neural Networks \(Sutskever et al., 2014\)](#)  
→ Encoder-Decoder 구조의 시초로, Transformer의 기본 구조적 아이디어와 연결됨.
- [Neural Machine Translation by Jointly Learning to Align and Translate \(Bahdanau et al., 2015\)](#)  
→ Attention mechanism을 도입한 대표 논문으로, Transformer의 핵심 개념과 직접적으로 연관 됨.

3. 기능소개


● Arxiv Agent

```
storage > paper > 1706.03762v7 > 1706.03762v7.md > # Keyword
1 # Attention Is All You Need
2
3 # Abstract
4 지배적인 sequence transduction 모델들은 encoder-decoder 구성에서 복잡한 recurrent 또는 convolutional neural network에 기반하고 있습니다.
5 가장 성능이 뛰어난 모델들은 attention mechanism을 통해 encoder와 decoder를 연결하기도 합니다.
6 우리는 recurrence와 convolution을 완전히 배제하고 오직 attention mechanism에만 기반한 새로운 간단한 네트워크 아키텍처인 Transformer를 제안합니다.
7 두 가지 기계 번역 과제에 대한 실험에서 이 모델들은 더 높은 품질을 보이면서도 병렬화가 더 용이하고 학습에 필요한 시간이 현저히 적다는 것을 보여줍니다.
8 우리의 모델은 WMT 2014 English-to-German 번역 과제에서 28.4 BLEU를 달성하여, 기존의 최고 결과(ensemble 포함)보다 2 BLEU 이상 향상시켰습니다.
9 WMT 2014 English-to-French 번역 과제에서는, 8개의 GPU로 3.5일간 학습한 후 41.8의 새로운 single-model state-of-the-art BLEU 점수를 기록했으며, 이는 문헌상의 최고 모델들에 비해 학습 비용이 매우 적은 수준입니다.
10 우리는 Transformer가 대규모 및 제한된 학습 데이터 모두에서 English constituency parsing에 성공적으로 적용됨을 보여줌으로써, 이 모델이 다른 과제에도 잘 일반화됨을 입증합니다.
11
12 # Summary
13 이 논문은 기존의 sequence transduction 모델들이 recurrent 또는 convolutional neural network에 의존하는 한계를 극복하기 위해, 오직 attention mechanism만을 사용하는 Transformer 아키텍처를 제안합니다.
14 Transformer는 recurrence와 convolution 없이도 높은 성능을 보이며, 병렬화가 용이해 학습 속도가 빠르고 효율적입니다.
15 기계 번역 실험(WMT 2014 English-to-German, English-to-French)에서 기존 최고 성능을 뛰어넘는 BLEU 점수를 기록하였고, 학습 비용도 크게 절감하였습니다.
16 또한, 영어 구문 분석(English constituency parsing) 등 다양한 자연어 처리 과제에도 잘 일반화됨을 보였습니다.
17 이로써 Transformer는 sequence modeling 분야의 새로운 표준이 되었으며, 다양한 데이터 크기와 과제에 효과적으로 적용될 수 있음을 입증하였습니다.
18
19 # Keyword
20 - Transformer
21 - Attention Mechanism
22 - Sequence Transduction
23 - Encoder-Decoder
24 - Machine Translation
25 - BLEU Score
26 - Parallelization
27 - Neural Network
28 - English Constituency Parsing
```


etary > storage > paper > 1706.03762v7
이름 ^
 1706.03762v7.md
 1706.03762v7.pdf


4. 한계

- 여러 개의 답변을 병렬로 답변할 수 없음.


 RUNNING... Stop Deploy


# Lang-Secretary




 <https://arxiv.org/abs/1706.03762> 다운 받아

Deploy

 안녕 우리집 강아지 이름은 철수야

 안녕! 우리집 강아지 이름이 철수인 거 기억하고 있어. 철수에 대해 궁금한 점이나 이야기하고 싶은 게 있으면 언제든지 말해줘!

 우리집 강아지 이름이 뭐라고?

requests.exceptions.JSONDecodeError: Expecting value: line 1 column 1 (char 0)

Traceback:

File "C:\Users\Xenrose\Desktop\Lang-Secretary\src\run\_streamlit.py", line 53,  
print(response.json())  
^^^^^^^^^^^^^^^^^^^^

File "C:\Users\Xenrose\Desktop\Lang-Secretary\venv\Lib\site-packages\requests\  
raise RequestsJSONDecodeError(e.msg, e.doc, e.pos)

## 4. 한계

- 지원하는 LLM api가 Chat-gpt 1개 뿐이며  
Agent/MCP tool을 calling해야 하기 때문에  
해당 기능이 없는 일부 LLM에 대해서는 사용이 불가

```
def generate_llm(model_company:str="openai", model_name:str="gpt-4.1", temperature:float=0.3, KEY:str=OPENAI_API_KEY):  
    ...  
    model_company: 사용할 모델의 회사  
    model_name: 사용할 모델의 이름  
    temperature: 모델의 온도  
    ...  
    if model_company == "openai":  
        return ChatOpenAI(model=model_name, temperature=temperature, openai_api_key=KEY)  
    else:  
        raise ValueError(f"Invalid model company: {model_company}")
```

Ctrl+L to chat, Ctrl+K to generate

## 4. 한계

- RAG 파이프라인의 부재  
(단, 필요성이 적다고 판단.)
- 예외 처리, Test code, 비동기, logging, monitoring 등  
프로덕트 기반에 대한 기능 미구현

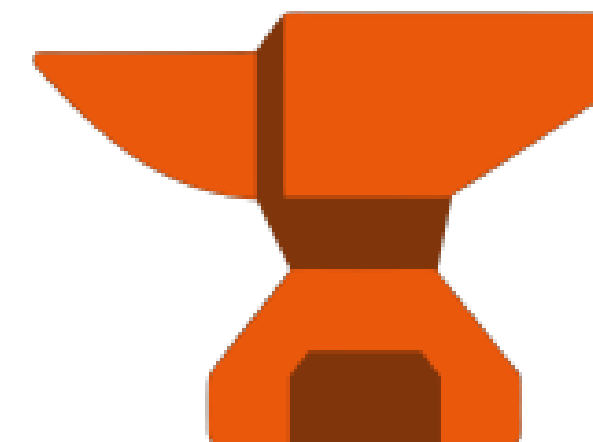
## 5. 추후 개발 사항

- 프로젝트 완성도 향상
  - pytest를 통한 test code 작성
  - LangSmith를 통한 LLM 답변 tracing



## 5. 추후 개발 사항

- 사용자 기능 향상
  - tts api를 활용하여 NotebookLM과 같은 논문 리뷰 팟캐스트 작성 Agent
  - 더 다양한 LLM api 지원
  - MCP server 자율적 추가



## 6. 후기

- LangGraph와 MCP를 통해 희망을 보았다.  
어쩌면 나도 다른 사람들을 도와줄 수 있는 프로젝트를 만들 수 있을지도 모르겠다.
- Ai 발전으로 인해 코딩 부분은 생각보다 할만 했다.  
하지만 프로젝트 아키텍처, test, logging 등  
엔지니어적인 요소에서 생각보다 고민이 많았다.
- 이후 업데이트는 아래 주소에서 확인 가능합니다.  
<https://github.com/Lt-kang>