by 김영욱 May 07. 2024

이 글을 읽으시기 전에 이전의 글 <10분만에 RAG 이해하기>를 먼저 읽고 오시면 지금 이 글의 이해에 도움이 됩니다.

0. 등장 배경

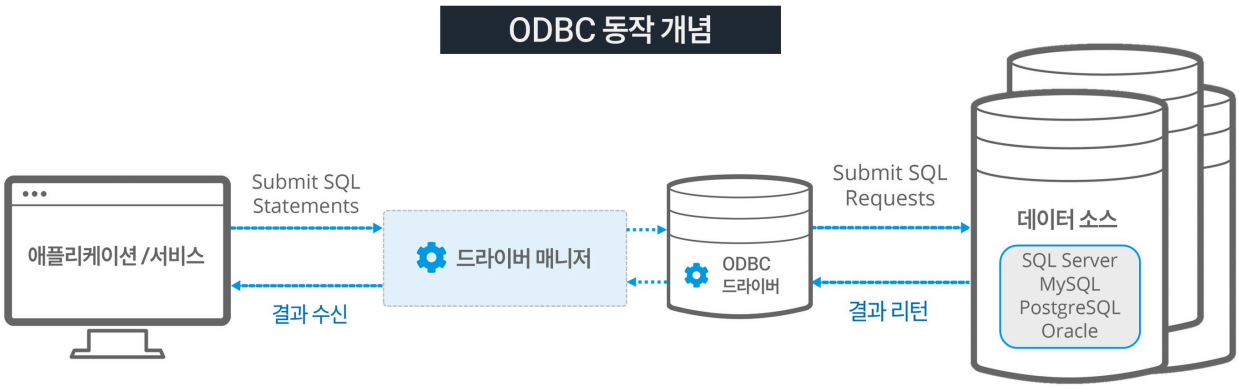
이미 잘 알고 계시지만, 이미 시장에는 OpenAI의 ChatGPT, 앤쓰로픽의 끌로드, 구글의 제미나이, 네이버의 하이퍼클로바X 와 같은 대표 LLM을 제외하고도 수많은 언어모델이 거의 매일 새롭게 나타나고 있으며 **각 모델마다 고유한 기능과 전문성을 갖추고 있습니다.** 그러다 보니 비즈니스 애플리케이션을 고집하지 않아도 어떤 서비스를 만들고자 할 때 그 기능 요구 사항에 따라 사용자 쿼리를 해석할 때는 어떤 특정 LLM을 사용하고, 해당 쿼리에 대한 응답을 작성하는 데는 완전히 다른 LLM을 사용하고자 하는 필요성이 생길 수 있습니다. 이럴 땐 어떻게 프로세스 파이프 라인을 구축해야 할까요? 이런 워크 프로세스를 요청하는 시나리오가 바로 랭체인(LangChain)이 탄생하게 된 아이디어를 제공합니다.

그런데 우리 본격적으로 랭체인 이야기를 시작 전에 잠깐만요.
저렇게 **여러 복수 시스템이나 백엔드 툴을 이용해야 하는 요구 사항이 발생하는 경우가 LLM이 처음일까요?** 그럴 리가요. 이런 경우는 아마 소프트웨어 탄생의 역사와 함께 아주 흔하게 일어납니다.

30년도 더 된 아주 오래되고 전통적인 예를 들어보겠습니다. 모든 백엔드 시스템에는 데이터를 저장하는 데이터베이스가 있습니다. 그리고 그 데이터베이스를 만들어 판매하는 기업들이 수없이 많았고요. 그런데 데이터베이스에도 각각의 특성과 장점이 있습니다. 오라클 데이터베이스가 탁월한 기능이 있겠지만, SQL

유지, 보수, 관리, 배포는 큰 오버헤드가 됩니다.

이런 시장의 요청이 **ODBC(Open Database Connectivity)와 JDBC(Java Database Connectivity)같이 데이터베이스 관리 시스템에 액세스 하기 위한 표준 API 레이어를 세상에 나오게 한 것**입니다. 이런 표준 API 레이어는 데이터베이스 시스템 및 운영 체제와는 독립적으로 동작하므로 ODBC/JDBC를 사용하여 작성된 애플리케이션은 데이터 액세스 코드를 거의 변경하지 않고도 클라이언트 및 서버 측의 다른 플랫폼으로 이식할 수 있는 장점이 있습니다.



ODBC 동작 원리

LangChain은 LLM을 사용하는 애플리케이션 개발을 위한 오픈 소스 프레임워크로, Python과 JavaScript 라이브러리를 제공합니다. 기본적으로 거의 모든 LLM을 위한 일반적인 인터페이스이므로 LLM 애플리케이션을 구축한 다음 통합할 수 있는 중앙 집중식 개발 환경을 갖추고 있습니다.

위에서 예로 설명한 ODBC가 동작하는 개념과 거의 똑같다고 보시면 됩니다.

랭체인은 2022년 10월 해리슨 체이스가 출시한 이후 GitHub에서 가장 빠르게 성장하는 오픈소스 프로젝트였습니다. 랭체인 안에는 무엇이 있는지 그 구성 요소를 살펴보도록 하겠습니다. 구성 요소 또한 무엇이 있다고 외우는게 아니겠지요? 어떤 필요성이 존재했는지를 알면 자동으로 구성요소는 머릿 속에 남아있을 거니 김PM 설명을 잘 따라오세요.

1.랭체인의 구성 요소

1) LLM 추상화(Abstraction)

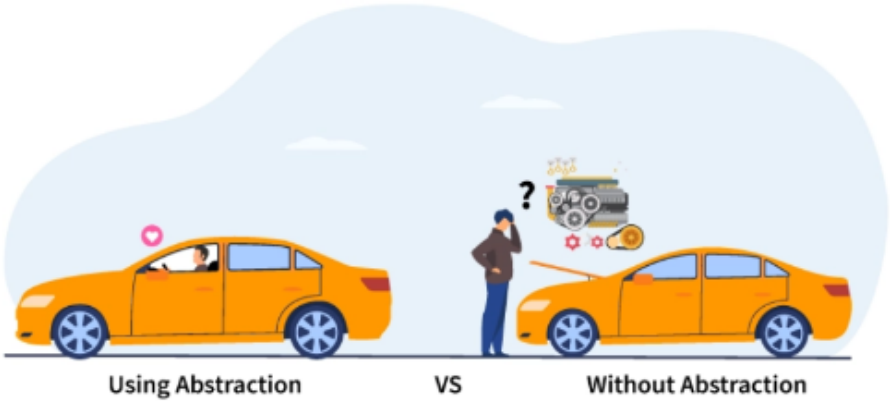
랭체인은 **추상화라는 것을 통해 LLM 애플리케이션의 프로그래밍을 간소화**합니다

수 있습니다. 예를 들어 설명하면 아주 쉽습니다.

여러분의 자동차가 추상화의 가장 좋은 예가 될 겁니다. 자동차는 키를 돌리거나 시동 버튼을 눌러 시동을 걸 수 있습니다. 엔진이 어떻게 동작하는지, 자동차의 다른 부품과 어떻게 연결되는지 알 필요가 없습니다. 자동차 내부 구현과 복잡한 로직은 사용자에게 완전히 숨겨져 있습니다. 여러분의 커피 머신도 추상화의 좋은 예입니다. 원래 한잔의 커피를 마시기 위해서는 물과 원두를 준비하고 원두를 가는 정도를 선택해야 합니다. 그런데, 커피 머신은 간단한 인터페이스를 통해, 이상적인 물의 온도나 분쇄 커피의 양을 알 필요 없이 신선한 커피를 내려줍니다. 커피머신의 내부 동작 원리를 알 필요가 없는 것입니다.

LangChain의 추상화는 언어 모델을 사용하는 데 필요한 일반적인 단계와 개념을 나타냅니다. 언어 모델(Language Model)을 연결(Chain)하여 애플리케이션을 만들 수 있으므로 복잡한 NLP 작업을 실행하는 데 필요한 코드의 양을 최소화할 수 있습니다. (Language Model + Chain = LangChain 이 된 거죠.)

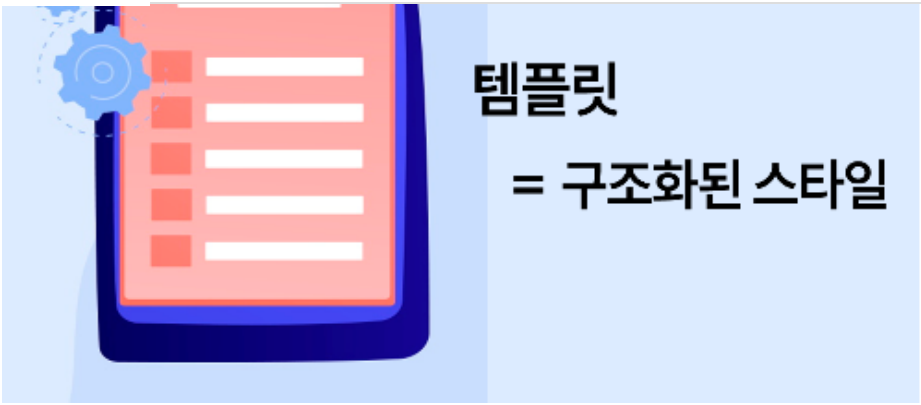
한 가지 언어모델을 이해하는 것도 충분히 어렵고 시간도 걸리는데, 여러 가지 언어모델을 모두 다 공부해야 한다면 미치고 팔짝 뛸 수도 있겠죠. 그걸 간단히 하자고 랭체인의 '추상화'가 도입된 것입니다.
거의 모든 LLM을 랭체인에서 사용할 수 있으며, 표준 인터페이스를 제공하도록 설계되었기에, 접속을 위한 API 키만 있으면 됩니다.



추상화는 사용자의 관점과 관련된 필수적인 세부 사항이 포함됩니다.

2) 프롬프트 (Prompts)

두 번째로는 프롬프트가 있습니다. 프롬프트는 LLM에 태스크를 수행하라고 전달하는 명령어입니다.
이건 여러분들도 ChatGPT나 그런 거 사용해 보셨을 테니 뭐 새로운 것도 없지요? ㅎㅎㅎ 그런 간단한 프롬프트라면 랭체인이 준비했을 리가 없죠.
LangChain에는 이 명령을 잘 전달하기 위한 **프롬프트 템플릿 클래스**가 있습니다. 여기서 중요한 말은 **'템플릿'**이죠. 우리가 문서 템플릿, 파워포인트 템플릿을 사용하는 이유는 무엇인가요? 바로 그 문서 내에 있



템플릿은 구조화된 스타일을 사용하기 위함이다.

예를 들어 그 템플릿 안에는 '응답에 전문 기술 용어를 사용하지 말고 답하세요'와 같은 지침이 포함될 수 있습니다. 또는 응답을 추가 설명하는 몇 개의 예를 같이 제공하라고 하는 즉 Few-Shot 프롬프트를 사용할 수도 있습니다. (프롬프트 엔지니어링 기술 중에는 응답에 예를 포함하라 마라를 지정할 수 있는 Zero-shot, One-Shot, Few-Shot 과 같은 종류가 있답니다.) 또한 출력 형식도 지정할 수 있습니다.

3) 체인 (Chains)

'체인' 말 그대로 연결 고리를 만드는 것이 랭체인 워크플로우의 핵심입니다.

체인은 LLM을 다른 구성 요소와 결합하여 일련의 태스크를 실행함으로써 애플리케이션을 생성합니다. 예를 들어 특정 웹사이트에서 데이터를 검색한 다음, 검색한 텍스트를 요약한 다음, 요약된 텍스트를 사용하여 사용자가 제출한 질문에 답해야 하는 애플리케이션이 있다고 가정해 봅시다.

이는 한 태스크의 출력이 다음 태스크의 입력으로 들어가는 순차적 체인이며, 체인의 각 태스크는 서로 다른 프롬프트나 심지어 서로 다른 LLM모델을 사용할 수도 있습니다. 다른 LLM을 사용하는 것은 1번에서 설명드렸듯 해당 LLM에 접속하기 위한 키 값만 있으면 가능합니다. 그런데, 모든 정보가 LLM에만 있는 것은 아니잖아요? 또한 LLM은 특성 날짜 이후에 업데이트된 데이터는 갖고 있지 않요. 저희가 이 이야기는 지난 글 <10분 만에 RAG 이해하기> 에서 RAG를 다루면서 말씀드렸던 부분인 것 다 기억하시죠? 그래서 랭체인에서는 바로 다음 4번의 인덱스 기능이 필요합니다. 그리고 밑에서 다시 한번 그림을 통해 설명드립니다.

4) 인덱스 (Indexes)

이제 특정 작업을 수행하기 위해 애플리케이션/서비스는 **자체의 학습 데이터 세트에 포함되지 않은 특정 외부 데이터 소스, 즉 내부 문서나 이메일 같은 것에 액세스**해야 할 수 있습니다. 그래야 **출처도 확실해지고, 보다 고급 전문 정보도 가져올 수 있고, 무엇보다 최신 업데이트 된 정보를 가져올 수 있습니다.** 랭체인에서는 이러한 외부 데이터를 총칭하여 인덱스라고 하는데, 그 인덱스 기능을 이루는 요소가 있습니다.

4-1 도큐먼트 로더 (Document Loaders)

첫 번째는 도큐먼트 로더입니다. 참고할 정보가 있는 파일을 읽어야 뭘 시작할 수 있겠지요? RAG 글 읽으

데이터베이스를 예로 들 수 있습니다.

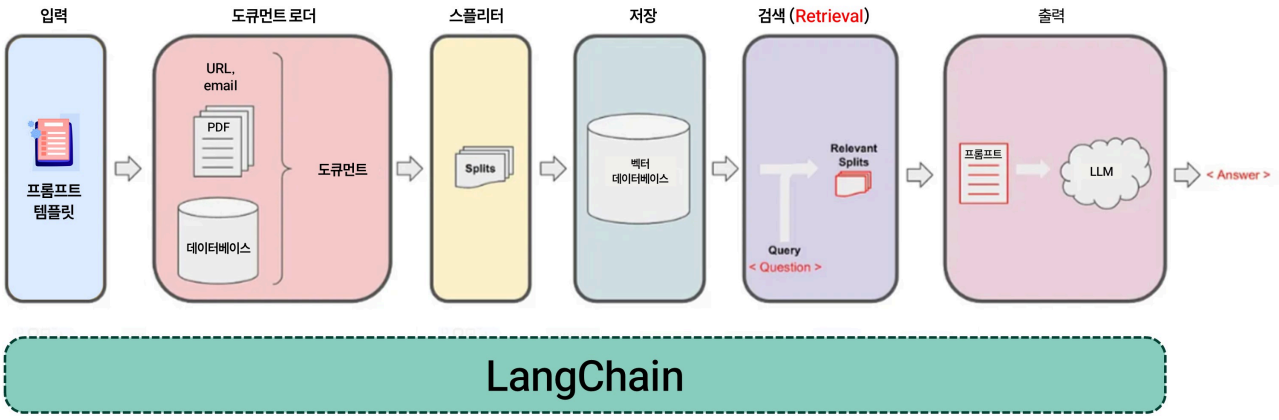
4-2 벡터 데이터베이스 (Vector Database)

벡터 데이터베이스에의 접근도 지원합니다. 기존의 관계형 데이터베이스와 달리, 벡터 데이터베이스는 데이터 포인트를 벡터 임베딩이라는 것으로 변환하여 표현하는데, 이는 고정된 수의 차원을 가진 벡터 형태의 수치 표현으로 유사성을 나타내기에 매우 효율적인 **검색(Retrieval) 수단**으로 사용됩니다. 제가 왜 검색을 search라고 이야기하지 않고, Retrieval이라고 기술했는지도 RAG글 읽으신 분들은 다 아시죠?

4-3 텍스트 스플리터 (Text Splitters)

텍스트를 원하는 방법과 매개변수를 사용하여 의미 있는 작은 덩어리로 분할/결합할 수 있는 텍스트 스플리터도 매우 유용합니다. 이렇게 분할 결합할 수 있어야 답을 줄 때 요약 정리를 깔끔하게 할 수 있겠지요.

한 번 이 부분을 RAG 워크프로세스에 대입해 보면 그림이 이렇게 그려질 듯합니다. 어렵지 않으시죠?

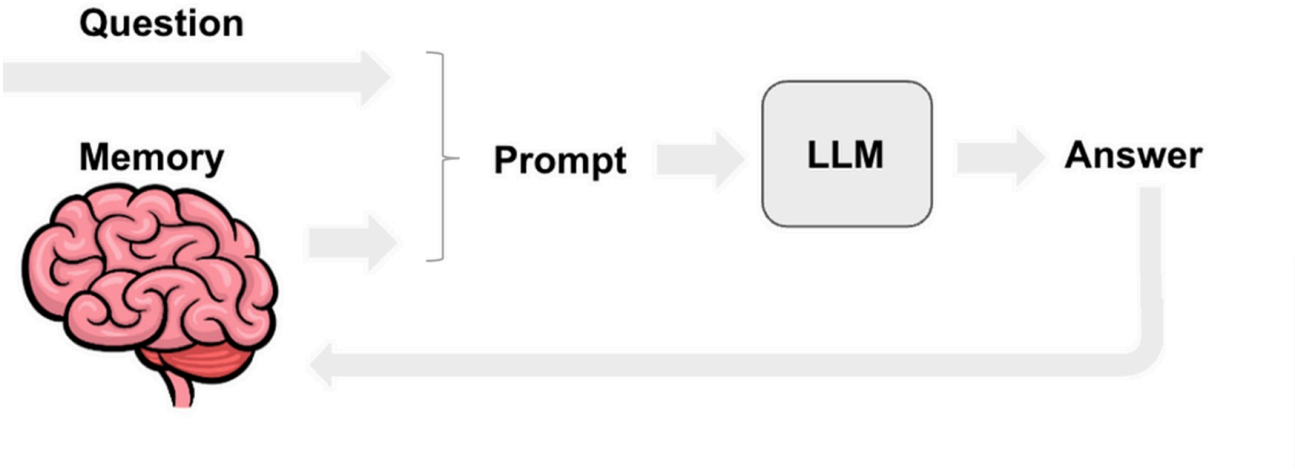


랭체인으로 구성한 RAG 워크 프로세스

5) 메모리 (Memory)

매우 중요한 기능입니다. 메모리란 사용자가 LLM과 프롬프트 기반의 '대화'를 하는 동안 사용자의 정보를 포함하여 대화에 대한 주요 사실을 기억하고 향후 상호 작용에 해당 정보를 적용할 수 있다는 뜻입니다. 즉 내가 누구이고, 대화의 문맥을 기억하고 이해하면서 대화를 한다는 뜻입니다. 이런 기능이 ChatGPT와 같은 인공지능을 훨씬 더 인간지능처럼 유용하게 만들 수도 있고, 스카이넷처럼 완전히 무섭게 만들 수도 있죠.

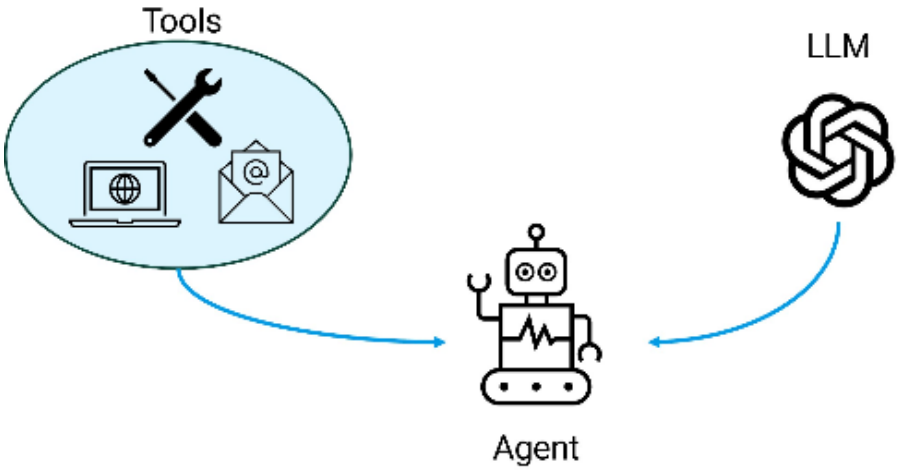
기본적으로 LLM은 입력으로 채팅 기록을 전달하지 않는 한 이전 대화에 대한 장기적인 메모리를 가지고 있지 않지만, LangChain은 애플리케이션에 메모리를 추가하는 방법으로 이 문제를 해결합니다. 두 가지 옵션을 제공하는데 지금까지의 **대화 전체를 기억하는 옵션과 지금까지의 대화 요약만 기억하는 요약 옵션**입



좀 더 똑똑해지는 기억력을 부여하는 메모리

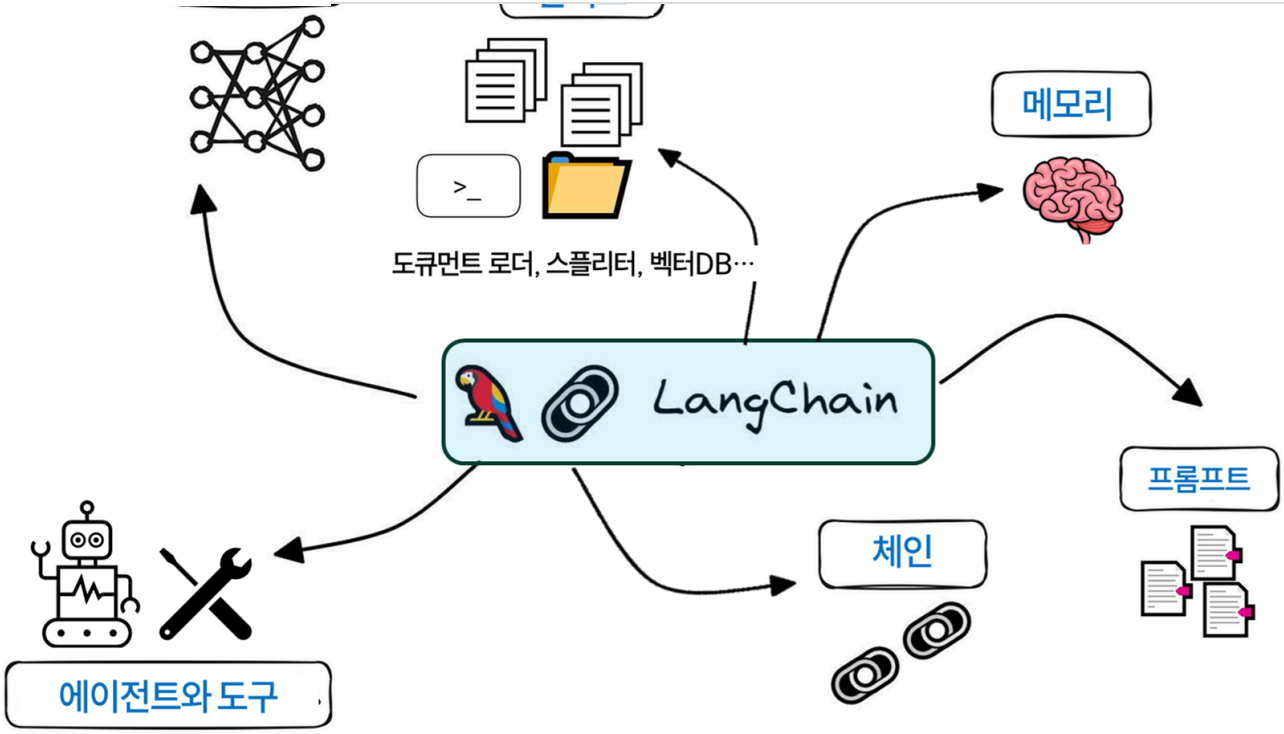
6) 에이전트 (Agents)

에이전트 서비스입니다. 가장 쉽게 이해할 수 있는 에이전트 서비스는 챗봇입니다. 많은 사이트에서, 많은 앱에서 이미 도입해서 사용하고 있습니다. 서비스 챗봇은 LLM에 있는 정보보다는 내부 정보, 고객 정보들에 훨씬 더 유용하게 동작해야 합니다. 랭체인의 에이전트는 LLM과 다른 데이터소스나 도구 두 가지 이상을 조합하여 사용이 가능합니다. **선택한 LLM을 추론 엔진으로 사용하여 어떤 작업을 수행할지 결정할 수 있습니다.** 에이전트를 구축할 때는 에이전트가 사용 가능한 도구 목록, 프롬프트 및 쿼리와 같은 사용자 입력, 이전에 실행된 기타 관련 단계 등을 입력으로 제공할 수 있습니다.



보다 개인화된 서비스 제공을 위해 여러 데이터 소스와 도구를 사용하는 에이전트

지금까지 랭체인의 구성 요소들을 알아보았습니다. 이해하는데 큰 어려움은 없으셨죠? 시간이 지남에 따라 이 구성요소들의 깊이가 깊어지고, 구성요소가 더 많아질 수도 있습니다. 그렇지만 늘 개념 정리가 잘 되어



랭체인 구성도

2. 랭체인으로 할 수 있는 일

랭체인의 구성요소를 살펴보았으니 이 구성요소를 갖고 할 수 있는 일을 생각해 볼까요? 이미 여러분은 다 알고 계실 텐데요.

1. 요약 기능

요약기능이 필요한 곳이라면 어디에나 적용 가능하겠죠. 프롬프트 템플릿을 사용하여 복잡한 학술 논문이나 자료를 분석하는 것부터 이메일의 요점만 제공하는 것까지 다양한 유형의 텍스트를 요약하는 작업을 수행할 수 있습니다.

2. 챗봇, Q&A

랭체인은 챗봇의 사용에 대한 적절한 컨텍스트를 제공하고 챗봇을 기존 커뮤니케이션 채널과 워크플로우에 통합하는 데 사용할 수 있습니다. 또한 특정 문서나 전문 지식을 기반으로 LLM은 저장소에서 관련 정보를 검색하고 유용한 답변을 제공할 수 있습니다. 거기에 맥락을 기억하는 메모리까지 이용하면 더욱 강력한 개인화 기능을 제공할 수 있을겁니다.

3. 데이터 증강(Augmentation)

RAG를 다루면서 다 이해하신 내용이죠? 전문 데이터를 이용하여 보다 정확하고 최신의 데이터를 제공하는

이 외에도 랭체인의 에이전트를 이용하여 다음 단계를 결정하여 진행하는 프로세스 오토메이션을 구축할 수도 있습니다. 랭체인은 오픈 소스이며 무료로 사용할 수 있습니다. REST API로 체인을 생성하기 위한 LangServe라는 모듈도 있고요. 애플리케이션을 모니터링, 평가, 디버깅하는 도구를 제공하는 LangSmith와 같은 프레임워크도 있습니다.

랭체인의 도구와 API는 LLM을 사용하는 애플리케이션을 구축하는 프로세스를 간소화시켜준다는 사실을 이해하셨으면 잘 소화하신 겁니다. 여기까지 읽고 이해하시느라 고생많으셨습니다. 또 다음 글에서 뵐게요.

AI 인공지능 스타트업

김영욱

커리어 분야 크리에이터

기획자

프로덕트 매니지먼트 저자

한국에서 학업을 마치고 7년간의 한국 후지쯔 근무후에, 프랑스 파리로 이주하여 현재 27년째 SAP의 프로덕트/프로그램 매니저로 근무중입니다. 책 <프로덕트 매니지먼트>의 저자.

구독자 2,637

매거진의 이전글 10분만에 RAG 이해하기

애플워치가 기능을 강조하는… 매거진의 다음글

디자인 독학하기 07 UI/UX 디자인 경험을 공유합니다 :) [Contents] UI 디자인을 위한 UX 원칙 10가지 01 우리는 사용자가 아니다. 02 각 화면에서 한 가지 행동에만…

by 김경환

이 글은... '이름 들으면 알 정도의' 스타트업이 아니라, 그 아래에서 성공하기 위해 고군분투하고 있는 작은 스타트업 주니어에 대한 글이다. 그리고 그 안에서 고군분투…

by 알토v

디자이너의 혼자서 하는 DIY공부법

디자이너의 혼자서 하는 DIY공부법 — Part 2. Part 1에서 말했던 것처럼, 나는 디자인의 정규 학교 과정을 거치지 않고서 바로 UI designer로 현장에 뛰어들었다. …

by 길리

18. 임베디드(Embedded) SW 개발자란?

HW에 내장된 SW를 개발하는 임베디드 개발자 전기자동차 시대가 되면서 자동차도 가전제품이 되었다 한다. 메카닉스가 집약된 대표 소비재였던 자동차가 전…

by OurStellar

챗 GPT로 자소서를 작성하면 안되는 이유

며칠 전 제게 자소서 컨설팅을 의뢰한 분이 있었습니다. 그 분의 자소서를 읽고 저는 다음과 같은 질문을 하였습니다. 자기소개서가 너무 추상적이네요. Chat GPT가 …

by 강동현 팀장

RAG or 파인튜닝? 선택 전 던져야할 몇가지 질문들

학습 차원에서 틈틈이 해외 전문가들이 블로그나 미디어 그리고 책에서 쓴 글을 번역 또는 요약 정리하고 있습니다. 이번 포스팅도 그중 하나고요. 거칠고 오역된 부분…

by delight

by 김영욱 May 07. 2024

매거진의 이전글 10분만에 RAG 이해하기
애플워치가 기능을 강조하는… 매거진의 다음글