**새로운 노트**

2025.05.27 화 오후 8:19 · 45분 10초
이태수

참석자 1 00:00
NLM이 검색하는 거 문서를 좀 보니까 사람들이 의미로 그냥 최신 거 보고 위키처럼 수작업으로 수정해 놓은 것 같더라고요.
그래서 막 라이브러리마다 수정 시간도 다르고 제가 직접 라이브러리 하나 골라가지고 아예 문서 프로포트 형식으로 올려놓을 수도 있고 그래가지고

참석자 1 00:32
MIT 라인센스 걸려 있어가지고 그냥 상관없어 그냥 쓸까 말까

참석자 2 00:42
이게 다큐멘트를 원하는 버전으로 줄 수 있지 않나 도큐멘트 내가 그거는 정확하게 저도 그 부분은 아는데

참석자 1 00:53
콘텍스트를 제가 봤던 거로는 최신 과 기준으로만 돼 있어가지고 뭐야 인턴이 끊겼네

참석자 2 01:09
버전이을 할 수 있는 걸가

참석자 2 01:15
4 제시 디스코도 그는 서 알겠습니다.

참석자 2 01:41
근데 이게 문서를 보면 버전 스페스픽 도큐멘테이션이라고 돼 있거든요.
특정 버전 지정이 가능할 것 같은데 온니 레이티스트 버전 이런 게 안 돼 있는 거 보니까 그래서 특정 버전을 명시하면 될 것 같은데요.
한번 그렇게 테스트를 해보시면 좀 좋을 것 같습니다.

참석자 1 02:01
일단 넥스트 JS 기준이긴 한데 컨텍 스피치의 분석 여기 이런 식으로 돼 있거든요.

참석자 2 02:11
버전이 있네요.

참석자 1 02:12
네 이거 넥스트 JS이긴 한데 13.5 14까지 사람들이 최신 것만 일단은 갱신하고 있다 보니까 비교적 최신 버전으로 가고

참석자 2 02:26
다른 거는 뭐 멋있어요? 혹시

참석자 1 02:30
이게 별거 다 있긴 하던데 테스트 API 테일 윈드 CSS 테스트 API는 버전이 이거는 진짜 최신 거 딱 하나만 있고 버전을 그렇게 자유롭게 지정하지 못할 것 같습니다.
여기서도 프린트가 다 없구나 플러스 뉴 버전 누르면 어떻게 돼요?
이거 이거 제가 적어야 되는 겁니다.

참석자 2 02:57
되는데 리퀘스트로

참석자 1 03:00
그거 요청을 하면은 요청 보고 요청 보고 적어주는 식으로

참석자 2 03:10
무조건 할 수 있는 게 아니고 진짜 랭 그래프나 이런 것들을 한번 참조를 해보면 좋을 것 같은데 그래서 요즘 나오는 이런 에이전트 프레임워크 같은 경우는 대부분 최신 버전이니까요.
그런 건 상관없을 것 같은데 파이썬 같은 경우도 한번 보고 자바도 한번 보고 해가지고 그러면 많을 것 같거든요.
파이썬은 어때요? 파이썬이

참석자 1 03:42
파이썬 그냥 파이썬

참석자 3 03:45
그걸 보면 더 좋을 것 같은데 파이썬 파이썬

참석자 1 03:53
최신 것만 있네요. 하나만 있어요. 네

참석자 1 04:09
그리고 a2 a

참석자 1 04:27
a 2a 식으로 구현해 보는 건

참석자 2 04:31
에이전트 테이전트요. 지금 그게 뭔가 필요하려면 제 생각에 저희들이 만든 이 HR 에이전트 자체를 를 다른 에이전트랑 또 송신하게 만드는 그런 시스템이면 뭔가 있어 보일 것 같은데 지금은 뭔가 하라면 할 수 있을 것 같은데 한번 그것도 일단 써보면 다 좋아요.
저는 무조건 해보면 좋다는 얘기긴 한데 그게 우리가 지금 하고 있는 거에 좀 잘 들어맞는 거면 상관이 없을 것 같고 따로따로 서버를 띄워서

참석자 1 05:08
연결하는 건가요?

참석자 2 05:11
그럴 거예요. 아마 그러면 그냥 냉그에 부가하세요.

참석자 2 05:24
아마 이거 될 것 같은데 컨텍스트 치

참석자 2 05:33
근데 다큐멘터리가 딱 거기 있는 것만 참고하는 거죠.
겟 라이브러리 덕스 이거 이거 가지고

참석자 1 05:39

보니까 이게 함수가 두 개 있더라고요.

참석자 2 05:43
걘 라이브러리 톡스랑 리주얼블 라이

참석자 1 05:45
쿼리로 임베딩 검색으로 라이브러리 찾는 거랑 그다음에 그거 찾은 거 아이디로 직접 까보는 거 문서

참석자 2 06:01
일단 한번 테스트 한번 해보세요. 거기 닥스 참고하는 거 말고 그냥 원래 파이썬이라든지 이 버전으로 한번 해봐달라고 한번 테스트해보고 그래서 한번 확인을 해봐야 될 것 같은데요.

참석자 1 06:18
제가 뭐냐 랭 그래프 도구를 다 df 그냥 동기식으로 선거를 했는데 예제를 보다 보니까 MCP를 도구로 선언할 때는 비동기로 선언을 하더라고요.
그래서 이게 같이 묶어 놓으니까 꼬이는 것 같아가지고 어

참석자 4 06:43
그거랑 페닝 팬 아웃이랑 관련이 있으려면

참석자 1 06:53
아니면 그냥 싹 다 MCP로 도구를 구현해 버린 다음에 같이 묶어버리는 거 생각은 한 번 해봤었는데

참석자 2 07:16
있어야 될 것 같아요. 에이 싱크로 df를 선언을 해야 될 것 같아요.
근데 그거는 뭐 크게 어렵진 않을 것 같은데 전부 도구 전부 a 싱크로 이렇게 호출하는 거 일단 함수만 그렇게 선언하고 그거 API만 받으면 되니까 그건 별 어려 비동기로 처리하는 게 나을 것 같아

참석자 1 07:40
이게 동기가 어쩔 때 좋을지 비동기가 어쩔 때 좋을지 아직 구분을 잘 못하겠어 가지고

참석자 2 07:46
그냥 뭐 사람들 많이 여러 툴 같은 거 호출하고 이럴 거면 비동기가 그래도 낫긴 하니까

참석자 2 07:58
일단 한

참석자 3 08:00
이제 보여드릴 것 중에 남은 거는 저희가 인호 님이랑 계속 이제 하루 매일매일 하루 한 시간 잡아서 스터디를 하고 있거든요.
스터디한 내용이랑 그리고 진짜 그게 어떤 식으로 코드를 구현하셨는지 그리고 그다음에 이제 시간이 남는다면 이제 저희가 금요일 날까지도 3주 차 산출물이 있단 말이죠.
그거에 대한 피드백을 받고 일단은 저희가 준비한 건 여기까지긴 해요.
네 일단은 저희가 첫 번째 스터디를 했을 때 지금 이 내용인데 기본적인 저희 팀이 사용한 랭 그래프에 대한 구조를 냉스믹스로 설계 한 거를 이제 작동 어떻게 작동하는지 보는 그런 스터디를 했었어요.

참석자 3 08:39
그래서 기본적인 랭 스미스를 실행하기 위한 그런 내용들이랑 그다음에 그렇게 해서 어떻게 되는지 이거는 스테이트를 구현하기 전 그러니까 슈퍼바이저 방식으로 저희 프로젝트가 어떻게 구현이 되는지에 대한 내용인데 여기 보시면은 이거는 항공이랑 플라이트 그러니까 호텔 예약 이거는 랭그래프에 있는 기본 예제로 나온 문서여가지고 저희 프로젝트 구조는 아니긴 해요.

참석자 1 09:09
랭 그래프 스튜디오로 하면은 좀 스터디 하기 편해 보여가지고

참석자 3 09:16
그리고 이게 오늘 한 내용인데

참석자 1 09:38
일단은 이거는 스테이트 그래프 형식으로 일단 구현을 했는데

참석자 2 09:45
그래서 이렇게 가 보여주면서 했구나.

참석자 3 09:49
네 실제로 각 에이전트가 그 안에 어떻게 돼 있는지 그래서 이런 식으로 일단 설명을 해 주시면서 했거든요.

참석자 1 10:02
이게 노드 하나 진행할 때마다 인터덕트 걸어가지고 중간에 멈추게 할 수도 있고 프롬프트나 툴 같은 거 수정하고 그 툴 선언하기 바로 직전부터 다시 시작하게 할 수 있는 기능 같은 게 너무 편해가지고 디버깅 같은 거 할 때

참석자 2 10:22
아 이

참석자 1 10:24
스튜디오 말하는 거예요. 이게 맥북에서는 로컬 도커로 실행 가능하던데 뭐냐 윈도우 환경에서는 그게 지원을 안 해가지고 랭스니스 API 키를 넣어서 스튜디오 베브 명령어 실행하면은 이런 식으로 이렇게 랭스미스 쪽 사이트에서 띄워주는 식으로

참석자 2 10:48
저는 이거 알았어.

참석자 3 10:49
실제로 툴들까지 구현을 한 다음에 그 랭 그래프 대

참석자 2 10:53
이거 다 들리고 이거는 랩 플로우랑 그게 다르구나 이거 코드로 한 거를 걸고 있네.

참석자 3 10:57
네 네 있습니다. 네네네네.

참석자 2 11:00
다 해놓은 거를 이렇게

참석자 1 11:02
객체만 마지막에 컴파일 해 놓으면은 보여주더라고요.

참석자 1 11:17
그리고 네

참석자 2 11:19
그래서 이제 그래프 방식 작동하는 것들 여러분

참석자 1 11:22
네네네. 그냥 슈퍼바이저 에이전트랑 니트 에이전트로 엣지 선언 안 하고 그냥 슈퍼바이저 에이전트 안에 리액트 에이전트 3개 넣어서 간단하게 구현한 거 하고 그다음에 이렇게 스테이트 방식으로 각각 에이전트들 노드 선언하고 엣지 연결해 주고 그런 거 비교해 가면서 일단 스터디

참석자 2 11:44
괜찮은데요. 이런 거 하시면 좋아요. 그래서 그래프 방식이랑 스테이트 그래프 방식 차이도 좀 보여주면 좋고 다 말씀하신 대로 크레이트 레드 엔저 같은 것도 보여주면 또 좋기 때문에 저는 이거 괜찮은 것 같아요.

참석자 1 11:57
네. 노드 엣지 스테이트 그래프를 한다고 해도 각각 에이전트 같은 경우에는 그냥 리액트로 선언하는 게 좋아 보이긴 하던데

참석자 2 12:12
사실 이 크리에이트 리액트 에이전트로 해도 상관은 없어요.
상관은 없는데 저번에도 말했듯이 이게 로깅화가 제대로 안 되거든요.

그래서 스테이트는 모든 스테이터를 다 추적해

참석자 1 12:24
볼 수가 있잖아요.

참석자 2 12:25
딕셔너리 형태로 보내가지고 그다음에 실제적으로 파이널틱 같은 걸로 이제 스트럭처 아웃풋도 다 정리도 해줄 수도 있고 그런 이제 세세한 컨트롤을 해도 할 수가 있으니까 아마 프로덕트 레벨에서는 그걸 할 수 있고 프레이트 에이전트 같은 경우는 이렇게 그런 MVP나 이런 조그마한 데모에서 사용하기는 좋다고 생각을 해요.
근데 사실 이게 필요 없는 호출을 또 난발을 할 때가 있고 또 LLM에 도 펑션 콜링에만 그냥 아예 의존을 하다 보니까 그거랑 프론포트만 의존을 하다 보니까 그런 부분에 대해서는 하시면서 만약 이게 에이전트 안에가 이제 툴들의 선언들은 더 길어지고 안에 폴백도 길어지고 이런 식으로 하면 좀 컨트롤 하는 부분이 그런 부분에 대해서 이제 아예 LLM 성능에만 맡길지 아니면 그거를 분기대로 내가 다 나눠 가지고 할지 그거의 차이기 때문에 여러 가지 세 개의 에이전트가 있으니까 거기서 몇 개는 그걸로 해보고 몇 개는 이걸로 해보고 이런 식으로 해도 상관없을 것 같아요.

참석자 3 13:29
강사님께서도 멀티 에이전트 방식 같은 거를 찾아보면 되게 여러 개 나온다고 하셔가지고 그중에서 고르든 아니면 직접 해본 다음에 하든 그래서 어떤 게 제일 좋았는지 얘는 왜 이거보다 좀 뭔가 성능이 좀 떨어지는 그런 거를 비교하면서 하면 더 좋다고 하시더라고요.

참석자 2 13:44
그래서 그래도 논리가 있으면 좋으니까 비교적 좀 작은 툴이나 이런 것들은 그

런 식으로 좀 호출하고

참석자 1 13:51
그래프 에이전트 그래프 멀티 에이전트 관련해서 공식 문서를 보면은 리액트 에이전트는 실제로 해볼 수 있게 자세하게 예제를 적어놓고 스테이트 그래프 멀티에이전트 항목은 그냥 코드 중간중간에 다 생략해놔가지고 바로 예제로 실행할 수 없게 대충 만들어 놨더라고요.

참석자 2 14:14
그 사람들이 예전에 그렇죠 그렇죠 그거를 그거는 왜냐하면 크레이티베이 센터는 그렇게 노드를 구현할 게 많이 없으니까 아마 그럴 거예요.
그건 또 툴들도 다 해야 되니까 그리고 아마 크레이트 레트 에이전트가 그게 안 될 거예요.
병렬 처리가 안 될 거예요. 네 그래서 다중으로 해야 될 때는 좀 힘들어서 이제 직선적 도구로 이제 아마 호출을 하는 거

참석자 1 14:41
변멸이라는 게 슈퍼바이저가 한꺼번에 에이저 2개

참석자 2 14:45
그래서 죽어야 하죠. 그런 걸 수도 있고 그다음에 뭐 툴 선택 같은 경우도 이제 우리는 분기 루프 조건 메모리가 엄청나게 가는데 이거는 그냥 선택 아니고 실행 이걸 반복만 하는 거잖아요.
그래서 그런 부분이 좀

참석자 1 15:00
그러면은 슈퍼바이저 쪽만 튜닝하면 되는 거 아니에요

참석자 2 15:03
어떤 슈퍼바이저 좀

참석자 1 15:05
슈퍼바이저 쪽을 리액트 에이전트로 리액트 에이전트나 뭐냐 슈퍼바이저 에이전트를 안 쓰고 노드 분기해 주는 거를 동시 에이전트 한꺼번에 2개 이상 호출할 수 있게 스퍼이저 쪽만 튜닝 해 주면

참석자 2 15:31
슈퍼바이저 자체가 이제 우리가 하려는 목적이 실제적으로 테스크들이 어떤 게 올라왔을 때 그 테스크에 대한 것들을 정확하게 판단을 하고 거기에 평가를 하는 거잖아요.
그래서 저는 이게 무조건 좋다 나쁘다 이런 건 없는 것 같고 그냥 뭐 아까도 말씀드렸듯이 코드로 이제 명시적 흐름을 설계해가지고 이걸 최적화시킬 건지 아니면 그냥 자연 프로포트 기반으로 그냥 행동만 유도를 할 건지 이걸로는 이것만 해도 상관이 없을 것 같은데로 판단해서 좀 의미를 좀 부여를 하면 좀 더 이제 이야기할 게 좀 더 많을 수 있을 것 같아요.
왜냐하면 이제 결과적으로 그런 것도 있을 수도 있어요.
실질적으로 어떤 에러가 나거나 디버깅할 때 스테이트 그래프 같은 거는 이제 코드 분기를 다시 짜거나 이런 부분을 하면 되는데 저기는 프로포티만 튜닝을 해야 돼요.

참석자 2 16:25
그래서 그런 부분에 대해서도 거기 안에서도 만약에 툴이 늘어난다고 보면 프로포트로 못 끝낼 수도 있고 좀 더 잡을 수 있고 하고 싶은 부분이 있는데 그런 부분이 좀 애매할 수가 있다.

참석자 1 16:36
그러면은 그 리액트 에이전트 라이브러리를 들어가서 까보면은 그것도 스테이트 그래프로 돼 있지 않습니까?
안쪽에서 아마 그럴 거예요. 그거를 그대로 가져와서 튜닝

참석자 2 16:49
그렇게 해도 상관없죠. 근데 그렇게 하면 결과적으로 에스테이트 그래프 아마 비슷할 거

참석자 1 16:58
그래서

참석자 1 17:19
그래서 일단은 에이전트 쪽 선언해 놓은 거 하고 풀 선환동 코드 분리해가지고 풀 쪽을 좀 업무 분담을 해가지고

참석자 2 17:33
네 그렇게 하면 제일 좋을 것 같고 잠깐 제가 이 코드를 보니까 그래 리액트 에이전트가 스테이트 그래프 기반이 아닌 것 같아요.

참석자 1 17:48
트는 인베드가 있고 SRC

참석자 2 17:53
아니고 이게 에이전트 익스큐터나 이런 거 같거든요.
에이전트 루프로 만들어져 있는 것 같은데

참석자 1 18:03
얘 밑에 결국에 컴파일 할 때 스테이트 그래프 안철 수가 있었어

참석자 1 18:31
스테트 그래프 맞는 것 같은데 안에 컨디셔널 엣지로 엄청 쪼개놓고 잠시만요.

참석자 1 19:05
선물은 함수마다 서브 그래프로 다 묶어놓고

참석자 2 19:20
라이브로 인식하셔

참석자 2 19:33
스테이트 그래프 아닌 것 같은데요. 아닌 것 같아요.
네 이거잖아요.

참석자 2 19:42
보시면 에이전트 분기로 그랬는데

참석자 1 19:47
이거 지금 이거 말씀하시는 거지 않습니까? 네네네.
이거 밑으로 내려보니까 안에 세세한 분기들을 누구야 지금 스테이트가 뭐지 했지

참석자 1 20:10
밑에 여기서부터 이제 툴 콜링하는 부분 같은 거는 툴 콜링은 좀 이렇게 돼 있

는 다 애드 지 에드 에드 했지 이런 식으로 밑에 가서 결국에 워크 플로우에 에드 애지 해놓은 것들 컴파일해가지고 리턴해주는 그런 식으로

참석자 2 20:30
그러네요. 그러네요. 근데 좀 애매하긴 하다. 저거를 굳이 뜯어가지고 하는 거랑 다를 게 없을 것 같아서

참석자 1 20:38
안에 세세한 프롬프트나 로직 같은 것만 만드는 식으로

참석자 1 20:49
프롱쿠트라고 해도 뭐가 별로 없네

참석자 2 20:51
없어요 없어 이거는 그냥 초콜릿만 하는 거지

참석자 1 21:11
강사님이 또 말씀해 주신 게 슈퍼바이저하고 에이전트를 선언해서 썼을 때 단점이 슈퍼바이저가 하위 에이전트한테 뭐 하라고 시키면은 하위 에이전트가 수행을 하고 수행한 데이터를 얘가 바로 사용자한테 전달할 수도 있는데 좀 슈퍼바이저한테 다시 전달해가지고 슈퍼바이저가 전달해서 사용자한테 전달하는 식이잖아요.
네 맞아요. 맞아요. 그거를 그냥 하위 에이전트에서 정보가 충분하다 싶으면은 그거를 사용자한테 바로 전달하는 좀 한 단계 간략화된 식으로 할 수도 있지 않냐 딜레이를 줄일 수 있는 레이턴시가

참석자 2 21:56

근데 이게 에이전트로 오면서 사실 그런 레이턴시 부분은 아직 해결이 될 수가 없는 부분이에요.
아시다시피 이런 크레이 에이전트도 그렇고 아니면 스테이트 그래프도 그렇고 결과적으로 툴 콜링을 하고 그걸 여러 번 우리가 개선을 해나가는 포백이나 이런 방식을 하는 거기 때문에 그게 있든 없든 시간이 엄청 크게 차이가 나지는 않을 거예요.

참석자 1 22:21
슈퍼바이저를 한 번 더 거치든 안 거치든

참석자 2 22:24
근데 그렇죠. 근데 그 거치 뭐 그거는 꼭 그럴 수가 없죠.
그러니까 데이터에 따라서 좀 다를 수도 있다고 생각을 하는데 사실 어떤 포맷을 쓴다든지 어떤 데이터의 할루시네이션을 검증한다든지 이런 스텝 자체가 필요한 부분이 있기 때문에 그래서 그것도 이제 상황에 따라서 쓰는 건데 사실 프로덕트 레벨에서는 이제 예를 들면 실제 우리가 신뢰를 많이 해야 되는 검색 엔진이나 이런 거에서는 편리할 수밖에 없죠.
불필요한 정보를

참석자 1 22:52
수가 있으니까

참석자 1 23:00
강사님께서 초 단위로 실행 시간이 늘어나면은 버리는 게 좋다고 그런 식으로 하셨어가지고

참석자 2 23:11

다 초단위 아닌가 근데 에이전트로 오면서 그거는 어쩔 수가 없는 부분이어서 그래서 요즘은 뭐 에이전트마다 슈퍼바이저를 달지 안 달지를 선택을 하잖아요. 전체 슈퍼바이저가 있을 수도 있겠지만 이 에이전트에만 슈퍼바이저를 달아 놓을 수도 있는 거고 아니면 상대적으로 아까 강사님이 말했듯이 조그마한 수의 작업이고 그냥 API 딱 건드려서 예를 들면 날씨만 추출하는 그런 간단한 거면 굳이 달 필요가 없고 그래서 그 슈퍼바이저는 어떻게 될지가 중요할 것 같은데 예를 들면 회사 조직도랑 똑같잖아요.
예를 들면 우리가 뭐 제가 지금 있는 게 클라이언트 엔지니어 팀인데 이 팀의 매니저가 이제 우리의 역할을 분담하고 보는 걸 수도 있는데 이 조직의 대빵의 직속에 있는 사람 리포터를 제출해서 조직이 바로 대표가 보는 사람은 그냥 이 이 대표가 그거일 수도 있거든요.
매니저일 수도 있잖아요. 그래서 그거는 또 상황에 따라서 슈퍼바이저를 어디에 달지를 선택을 하는 문제일 것 같아요.

참석자 2 24:05
그래서 경사님이 해준 포인트로 어떤 에이전트는 달고 어떤 에이전트는 안 달고 이런 식으로 나가면 차라리 좀 더 리즈너블 할 수도 있을 것 같아요.

참석자 2 24:22
그리고 대충 생각

참석자 1 24:25
모든 에이전트에 앤드 노드 단으로 가는 툴을 다 쓰게 해줘가지고 자기가 끝 맞힐 수 있다 판단하면은 바로 엔드 노드로 가는 툴을 호출해서

참석자 2 24:40
그럴 수도 있겠죠. 그것도 하나의 라우터가 되겠죠.
근데 그거를 판단하는 것도 어떻게 보면 하나의 마지막 결과를 검증을 하는 거

고 한 번 노출하는 거 똑같이 근데 그거를 에이전트에 대해서 어떻게 큰 예를 들면 이런 식도 있을 것 같아요.
예를 들면 작은 모델들을 여러 번 호출을 하잖아요.
그럼 이 답변에 대해서 만약에 실제로 프로덕트 레벨로 가면 작은 모델들을 써야 되는 사람밖에 없으니까 거기 나온 결과를 검증하지 못할 때는 꼭 큰 체트pd 같이 400불리어 넘는 그런 모델의 에이전트를 붙여야 될 수도 있는 거고 더 풍부하게 만들어 낼 수도 있는 거고 그런 부분이 이제 필요한 부분이어서 그것도 약간 짜기 나름인데 저도 저번에도 말씀드렸지만 에이전트 1 2 3을 다 똑같이 안 짜고 MCP도 붙여보고 방금 말했던 램체인으로도 붙여보고 그런 자원 리소스에 맞게 좀 설계를 해 주시면 사실 썰 풀 건 엄청 많아질 것 같아요.

참석자 2 25:42
좋은 포인트인 것 같아요. 그러 그래서 그런 식으로 같이 스터디 하면서 한번 구현을 해보면 제일 좋지 않나라고 생각이 듭니다.
사실 실제로 아까도 정원님한테 말했지만 대기업이랑 이제 스타트업들 있잖아요.
그러면 대기업들에서 은근히 이런 걸 잘 모르는 사람이 더 많아요.
스타트업의 사람들이 더 잘 아시는 분들이거든요.
거기는 빠르게 빠르게 이제 투자를 받고 하려면 신기술을 많이 도입을 하고 그런 걸로 이제 정부 과제도 따고 이래야 되는데 대기업에 하는 사람들은 지금 벌써 서비스가 나와 있고 돈을 벌기 때문에 그 서비스만 하고 있거든요.
그래서 다시 뭔가 공부를 하고 하니까 좀 힘들어 하시는 분들도 있고 아예 거기서 나고 되시는 분도 있고 막 그렇거든요.
그래서 지금 하시는 이런 단계들을 모르는 사람들이 많아요.
그래서 이런 걸 잘 해가지고 스터디에서 하면 면접 때 막 풀잖아요.
뭔 말인지 모르는 사람들이 많아요.

참석자 2 26:38

근데 이제 이걸 가지고 프로덕트를 만드신 분들은 그런 걸 다 이제 검증을 해 줘 그러면 너희들 멀티턴 대화 같은 거 할 때 이전 대화 같은 것들 관리하는 거 캐신 같은 거 했냐 이런 것들을 해서 다 들어오는 거죠.
그건 어떻게 했느냐에 따라 이어지고 프로트 랩에서 넣는 게 중요한 거니까요.

참석자 1 26:53
저희는 메시지 뱉을 때마다 저희 포스터 그리 세션 쪽 메시지 테이블에 다 저장해 놓고 그 뭐냐 장고 쪽에서 불러와서 넥스트 as로 넘겨주는 그런 식으로

참석자 2 27:13
해놨어요. 그래서 비슷한 것들이 있을 때면 그런 비슷한 답변을 내주게 하는 그런 표시 작업 같은 것도 해놓은 거예요.

참석자 1 27:20
그러니까 테이블을 만

참석자 2 27:21
있으면 너무 좋아요. 그런 것들 해놓으면 진짜 좋아요.

참석자 1 27:27
그럼 저 erd

참석자 3 27:29
네 이 아이디 이 아이디 보여드릴게요.

참석자 1 27:32

저기 erd 쪽을 이번에 아까 이용자 수 같은 거나 그런 거 데이터를 SQL을 테스트 SQL로 불러오려면 일단 포스터 그리 쪽에 넣어놓는 게 낫다고 생각을 해가지고 테이블을 3개 정도 밑에다가 더 만들어 봐야 되나 저희 일단 저희 위에 거는 저번에도 봤어 가지고

참석자 3 28:02
여기는 보여드렸던 거죠. 저희 만든 거는

참석자 1 28:06
이거 3개 아까 그 뉴스 기사하고 뭐냐 이용자 현황 그리고

참석자 3 28:17
가입 가입 이

참석자 1 28:18
가입 이탈 일단은 테이블 만들어 놓고 에이전트가 SQL 에이전트든지 아니면 저희가 허리를 내뱉는 프롬프트를 좀 넣어줘가지고 할지는 아직 안 정해 기능

참석자 2 28:34
지금 이거는 그 텍스트 SK 말씀하시는 거예요.

참석자 1 28:37
네 이거 텍스트 SQL로 불러오려고 일단은 포스터 그리에다가 저희 데이터 일단 넣어놓고 테이블 만들어서 이런 식으로 넣어놨거든요.
넣어 놓을 예정입니다. 일단 테이블은 만들어 놨고

참석자 2 28:52
일단 아까 그런 캐싱 같은 것도 이 멀티폰 대화 전체를 잘 저장을 해가지고 비슷한 흐름을 판단했을 때 비슷한 대답을 불러오게 하면 이런 비용 효율적에서도 또 이야기를 할 수 있는 부분이어서 저는 그런 건 좋은 것 같고 텍스트 q SQR도 꼭 하나가 있으면 좋긴 할 것 같아요.
그것도 회사에서 관심이 많은 부분이 근데 사실 회사 레벨에서는 텍스트 스퀘어를 어떤 걸 관심이 많냐면 이런 지금 컬럼들을 전체 컬럼 인리치먼트라고 부르거든요.
컬럼에 대한 디스크립션을 그냥 대략 테이블만 보고 자동으로 LM이 다 달아줘요.
그러면 그 LRM이 그걸 판단해서 그걸 보고 텍스트 SQL을 만들 거잖아요.

참석자 2 29:32
근데 테이블 이렇게 지금처럼 한두 개 이렇게 있을 때 한 3개 정도 있을 때 그거 보고 하는 거는 그냥 기본 하나의 LLM으로도 할 수 있는데 실제 대화 회사 레벨로 보면 이렇게 만 개 20만 개의 조인이 다 돼 있단 말이에요.
그거에 대해서 그 관련성을 찾아가지고 텍스트 스케어를 하는 거기에 관심이 제일 많거든요.
해본 경험도 중요하기 때문에 그래서 여러 가지 조인이 돼 있는 그런 sq 테이블에 텍스트 SQL을 한번 진행을 해 보시면 너무 많이는 아니어도 한 그래도 네다섯 개라도 해보시면 좀 그래도 많이 도움이 될 것 같아요.

참석자 1 30:08
제가 저번에 해본 토익 프로젝트 때 SQL 쪽으로 해본 거는 각 상황에 맞는 데이터 불러오는 SQL 쿼리를 함수로 툴로 지정을 해놓고 각 상황에마다 툴로 써가지고 정해진 쿼리 날려가지고 불러오는 식으로 일단 구현을 한번 해봤었고 제가 이렇게 구현을 한 이유가 맨 처음에 그냥 SQL 에이전트 박아놓고 유저 커리에 맞는 SQL 구분 짜가지고 알아서 해라 그런 식으로 했다가 딜레이가 너무 길

어가지고

참석자 2 30:49
딜레이가 길었다는 게 그것도 하나의 크리트 레이테이전트처럼 불러

참석자 1 30:53
SQL 에이전트가 일단 얘가 알아서 저희 SQL DB 스케마를 읽고 스케마 각각 컬럼이 뭐가 있는지 판단을 하고 그다음에 또 유저 커리에 맞는 SK 구을 생성해서 날리는 시기여가지고 좀 딜레이가 많이 걸려서 그냥 각각 상황에 맞는 거 중간에 셀렉트 그쪽 부분만 비워놓고 알아서 그 부분을 채우게 하고 그냥 각각 상황에 정해진 툴을 정의해 놓고 자세한 스케마는 프롬프트에다가 박아서 내가 알아서 상황에 맞는 커리 좀 딜레이 빠르게 날릴 수 있게

참석자 2 31:43
근데 이게 스키마도 나중에 보면 결과적으로 레그처럼 들고 오는 게 나올 수도 있어요.
레그처럼 왜냐하면 스키마가 지금은 이렇지만 아까 뭐 이상 탐지 이런 데 가면 센서 데이터 컬럼만 해도 100만 개가 이렇거든요.
그 스키마 자체를 II을 프로필에 다 담을 수가 없잖아요.
그렇죠 그래서 그런 부분도 좀 고민을 그거는 또 엄청 중요한 포인트예요.
회사에서 엄청 관심이 많은 그래서 그런 부분으로 이제 해보는 것도 좋아요.
크롬포트만으로는 이제 해결하기 힘드니까 그걸 디스크립션만 잘해 놓으면 디스크립션에 어떤 테이블에 종속돼 있고 이런 것만 잘 돼 놓으면 그런 텍스트의 SQL을 좀 어느 정도 해결할 수 있고 텍스트의 SQL도 어떻게 보면 밸리데이션 하는 그런 테스트 셋들이 많거든요.
그래서 그런 것들도 한번 한번 테스트해 보시면 더 좋을 것 같아요.
좋은 포인트인 것 같아요.

참석자 1 32:34
이것도 상황에 맞는 SQL 구먼들 다 만들어서 풀로 해놓는 게 나을까요?
아니면은 그냥 자연어로 알아서 그냥 SQL 구문 처음부터 생성하게 하는

참석자 2 32:43
근데 상황에 맞는 거라는 게 풀샷으로 이거는 이 정도에서 이것만 너 빈칸 채워 이런 거 말씀하시는 거예요.

참석자 1 32:50
시나리오 생성할 수 있는 건 다 해가지고

참석자 2 32:53
근데 그거는 좀 한계가 있을 것 같아요. 예를 들면 진짜 더 이상 이제 우리가 좀 노후화된 공장이고 여기 한 10년만 더 돌리면 된다 그런 거면 딱 정해진 센서가 더 들어올 것도 없고 그런 거면 그렇게 그냥 완전 오모 피팅을 맞춰가지고 거기만 딱 되게 만들어 놓으면 상관이 없거든요.
근데 앞으로 어떤 일이 일어날지 모르고 프로포트에 뭐가 하나만 더 추가가 되고 이렇게 하면 성능이 갑자기 또 뒤틀릴 수가 있고 이러기 때문에 사실 법령적으로 제일 잘 되는 게 제일 좋은 것 같아요.
그런 지금 그런 고민들도 많이 합니다. 그래서 좋은 포인트인 것 같습니다.

참석자 1 33:31
근데 이런 좀 비정형화된 데이터들이 막 여러 개 있으면은 몽고 DB가 좀 하고 싶기도 하고 그런 느낌이 들 그건 낭비겠지만 그냥 s3 그런 생각을 그냥 s3에다가 놓는 게 더 낫겠

참석자 2 33:52

그냥 클라우드 오브젝트 스토리지를 활용해서 하는 경우가 제일 많고요.
왜냐하면 그걸로도 대부분의 LLM이 이제 처리가 되니까 몽고 디비케이스는 아직 많이 못 봤어요.
못 봤었구나 아직

참석자 1 34:07
이게 뭐냐 벡터까지 지원을 하길래 원거리가 비정형까지 넣어주고 그래서 한번 생각을 해봤었습니다.

참석자 2 34:24
그것도 뭔가 목적에 따라서 좀 달라질 것 같긴 해

참석자 2 34:32
뭔가 실시간 조회 이런 부분이면 방금 말씀하신 게 맞을 것 같고 그게 아니고 이제 파일을 받아가지고 배치 잡을 한다든지 그런 처리 지금은 뭔가 실시간성이 엄청나게 있는 건 아니니까요.
써보면 써봤다고 하면 좋을 것 같은데 그런 빠른 처리를 뭔가 하려고 하면 몽고 DB를 좀 쓰는 것도 좋다라고 생각을

참석자 2 35:05
멀티턴 대화 저장 같은 경우는 몽고 DB 해도 되겠네요.
그런 게 더 빠르긴 하니까

참석자 1 35:14
그 아까 제이슨

참석자 2 35:17

네 그러면 이제 아까 말했던 그런 캐신 같아

참석자 4 35:21
그것도 할 수 있으니까

참석자 1 35:23
근데 아예 세션 그러면은 포스터 그리도 제이슨으로

참석자 2 35:34
8일 중에

참석자 1 35:35
넣으면 되지 않습니까?

참석자 2 35:36
뭐가 더 빠른지는 좀 테스트를 해보면 좋을 것 같은데

참석자 2 35:53
데이터는 그러면 이제 아까 그거

참석자 3 35:56
네네 이 데이터는 이제 저희가 찾으려고 하는 데이터니까 다 주시게 되는 거고 벡터 디베이트 사내 문서 같은 건 벡터 디베이트 파인 콘 디베이로 들어갔고 나머지는 저희가 아직 DB에는 안 넣었지만 이제

참석자 1 36:10

사례 분석 원본 같은 거 일단은 쓸 수가 있을 것 같아 가지고 에이스 3에다가 일단 원본을 올려놓고

참석자 3 36:19
아이폰을 집어넣으신 거 아니

참석자 1 36:21
하인폰에는 텍스트랑 벡터가 있긴 한데 이게 정제 전처리한 데이터다 보니까 네 사용자가 원본 사내 문서 같은 걸 원할 수 있는 요청을 할 수 있죠

참석자 3 36:33
맞아요. 네네네

참석자 2 36:36
그러면 데이터는 다 수집이 된 건가요? 대부분 그럼 이제 그냥 이제 하나씩 하나씩 에이전트 맡아가지고 RNR 정해서 이제 하면 되겠네요.
그러면 대략적으로 정했나요? 어떻게 될지 전체 큰 흐름은 인호 님이

참석자 1 36:51
제가 일단 툴 쪽은 에이전트 쪽이 어떻게 변하든 일단 다 쓸 것 같아가지고 각자 일단 툴을 배분을 하려고 했는데 그 에이전트별로 일단 코드 에이전트 같은 경우에는 제일 후순위로 해놓고 2 3 그러니까 예측하고 레그 쪽에 레그 쪽이 난이도가 좀 낮다 보니까 그쪽부터 저희가 맡아서 다 완성을 시킨 다음에

참석자 2 37:21
네 다 빠르게 끝낼 수 있는 것부터 그런 것들을 배분을 해서 실질적으로 해보게 다 하세요.

직접 다 해보시는 게 제일 좋을 것 같고 하나씩 마자가 같아가지고 지금 뭐 하나의 에이전트에도 툴이 거의 3개씩 붙어 있으니까 그래서 그 툴부터 하나씩 작업해서 패스트 API든지 아니면 MCP라든지 어떻게 말지 좀 정해주시고 그런 것도 가르쳐 주시고 하면 좀 좋을 것 같은데요.

참석자 1 37:47
그래 스튜디오 쪽에서 에이전트가 좀 완성됐다 싶으면은 패스트 API로든지 다른 거로든지 해가지고 장고 쪽에다 연결해서 테스트해 보고 있습니다.

참석자 2 38:01
그렇게 해도 좋죠

참석자 1 38:05
이게 좀 제 생각이 많은 게 캔버스 띄우는 거 있잖아요 네 캔버스 만약 저희가 이거 랭 그래프를 API로 만들어 가지고 장고 쪽이랑 연결을 했을 때 캔버스를 띄울 때와 안 띄울 때나 캠퍼스 종류 같은 거 코드 같은 거면은 코드 캔버스 보여주고 뭐냐 레그 에이전트 같은 경우에는 보고서 같은 거 NLM이 정리해서 캔버스 띄워주고 그런 거를 좀 어떻게 구분을 해서 해줄지 API로 불러왔을 때 그게 잘 설계가 안 돼가지고 캔버스 호출이랑 그 캔버스 종류 같은 거 일단 에이전트 무슨 에이전트 쓰는지 그거를 메시지 마지막에 해가지고 그걸로 하면 될 것 같아

참석자 2 39:04
그럴 수도 있고 아니면 리턴 값을 이제 스테이트로 받아가지고 그거에 이제 대화 맥락 같은 거 포함돼 있으면 그걸로 이제 캔버스 같은 거 열어줘도 좋을 것 같은데 방금 말씀하신 대로 에이전트별로 캔버스를 다르게 가는 것도 나쁘지 않을 것 같고요.
판은 다 똑같을 거니까 캔버스를 어떻게

참석자 1 39:26
띄울지만 설계가 안 돼요 어떻게 띄울지가 어떻게 띄울지가 랭그래프 단 쪽에서 캔버스 띄우라고 툴을 툴을 호출해서 캔버스 띄우라는 신호를 장고 쪽에 보내서 수신을 해서 할지 어

참석자 1 39:53
그것도 방법일 것 같고요. 아예 그 캔버스 캔버스 그리는 툴을 그냥 해버려서

참석자 2 40:03
그러니까 그것도 이제 상황에 맞게 이제 띄우려고 하는 거죠.

참석자 2 40:10
뭔가 이것도 스테이트 기반으로 가야 될 것 같은데 스테이트 기반으로

참석자 2 40:23
이제 얘기하고 서

참석자 1 40:24
생각 단어 말고 LLM 리스폰스 검증에서 사이에 땡땡땡 캔버스하고 캔버스 내용하고 땡땡땡 있으면은 그거 캔버스로 처리하는 그런 거밖에

참석자 2 40:38
그것도 괜찮죠 사실 지금도 사실 저번에도 말씀드렸듯이 실제 검색 서비스에서는 딕셔널 같은 페어를 엄청 많이 만들어요.
그런 하드 코딩 방식도 많거든요. 그래서 어떤 것들 특정 요약해줘 아니면 어떤

결과물 만들어줘 아니면 코드 만들어줘 같이 그런 것들을 이제 요청이 왔을 때 그거를 이제 하드 코딩도 몇 개를 만들어 놓고 방금 하는 그런 분기철도 만들어 가지고 두 개를 동시에 했을 때 그때 캔버스가 뜨게 하는 게 나을 것 같거든요. 그래서 그런 식으로 됐을 때 캔버스 응답 노드만 따로 만들어 가지고 글로 가지고 응답이 나오게 그런 식으로 하면 될 것 같습니다.

참석자 1 41:26
그리고 코드 컨버전 같은 경우에는 콘텍스트 7 같은 걸로 추가 정보를 준 다음에 lna 깡 성능에 맡기는 게 나을까요?
아

참석자 2 41:40
그도 코드 컨버전을 우리가 여러 개를 했었잖아요.
컨버전을 할지 도큐멘테이션만 보고 할지 여러 개를 했었는데 일단 컨텍스트 7 같은 경우가 웬만큼 다 나오잖아요.
그것도 이제 솔로 73을 다 반영해서 말아놓은 거니까 별로 변환 없이 그냥 올리는 게 제일 나을 것 같긴 한데 변환 그냥 거기 LNN 깡 성분에 LNN 근데 그거를 이제 아까 말했듯이 뭔가 또 검증을 할 수도 있는 부분도 있을 거고 사실상 근데 이게 어떤 깊이로 가냐에 따라 다를 것 같아요.
우리가 지금 할 거는 전체 폴더 구조를 다 봐가지고 모든 폴더 구조를 다 바꿔주고 이런 건 아니잖아요.
그래서 한 판이니까 그래서 그 정도는 괜찮지 않을까라는 생각이 드네요.

참석자 1 42:29
그냥 GPT 오픈 AI가 코딩에 맞게 튜닝했다는 GPT 4.5를 쓸지 아니면은

참석자 2 42:39
그것도 이제 모델 셀렉션을 이제 어떻게 평가할지를 정성 정량적으로 좀 평가

를 해보면 좋을 것 같아

참석자 1 42:44
아니면 럼팟에서 GPT 4.5보다 성능 좋다는 데브스트랄 불러와가지고 14위짜리 불러와서 할지

참석자 2 42:53
그것도 제 생각에 여러 가지를 좀 테스트를 한번 해보면 좋을 것 같은데 사실 면접에서 가장 많이 보는 게 이거를 왜 선택했냐 근데 그게 최근에 벤치마크에서 제일 높아 사유는 안 통해요.
그게 우리한테 높일지 너희가 어떻게 하냐 이런 식으로 나오거든요.
그래서 항상 이 테스트 셋을 만들고 아까 말했듯이 여러분들 5명이니까 5명이서 그 코드에 대해서 점수를 줘서 정성적 평가 척도도 만들어 놓는 게 좋아요.
그래서 발표하실 때 그런 거 보여주시면 좋아하거든요.
그래서 우리는 테스트 셋에서도 이게 전체에서 이런 점수가 나왔고 정성적 평가했을 때도 5명이 돌아가면서 이걸 평가했기 때문에 이 테스트 베이스를 만들어놓고 모델을 하나씩 하나씩 해가면서 테스트를 해봤다.
그래서 이 점수가 제일 잘 나오는데 이걸 뽑았다 이런 식으로 가는 게 좋을

참석자 1 43:35
평가 기준이 좀 객관적인 기준이 있어요.

참석자 2 43:38
그거를 이제 크리테리어를 좀 만들어 가야죠. 그래서 기계적 평가를 할 때 이 코드 평가하는 그게 있거든요.
그걸 그것도 하나 쓰고 정상도 평가는 이제 여러분 인사이트들이 다 다르잖아요.
이런 것들도 엄청 좋은 척도예요. 그래서 이런 것도 하고 예를 들면 라마 2가 처

음 나왔을 때 가장 떴던 게 뭐냐 하면 라마가 나오기 전에 이제 사람들이 말을 했던 건 파인 튜닝을 할 때 엄청나게 많은 데이터를 넣어야 된다.
그래 아케 하지 않으면 성능이 안 나온다고 했는데 그때 리마라는 논문이 나왔어요.
그 데이터 3천 개로 라마 2를 이제 만든 거였거든요.
알파카라 그래서 그게 이렇게 해도 성능이 좋아진다.
이 3천 개는 케냐에 있는 어떤 전문 요원들을 한 몇백 명을 투입을 해가지고 이 데이터를 3천 개를 만든 거예요.

참석자 2 44:23
그 3천 개도 그 사람들 혼자만 만든 게 아니고 얘가 만들면 얘가 검증하고 이렇게 하면서 만들어 그래서 사람의 정성적 평가의 인사이트도 엄청 높기 때문에 각각의 점수를 주고 그걸 평균을 매긴다든지 그런 식으로 하는 평가도 중요합니다.

참석자 1 44:48
드셨나요? 이제

참석자 3 44:50
질문 아니 그게 아니라 인호 님이랑 멘토링 지리 공장 서 기다려보

참석자 2 44:56
미쳤어 끊을 수가 없어 화장실 한번 갔다 올게요. 아니 잠깐 하나 씨 밀물이 어디 있지.