권구민 지원자에 대하여

권구민 인적사항

나이 : 만 20세, 2003년 05월 16일 생

성별 : 남자

LLM Engineer로 취업/이직을 하고싶어합니다.

연락처

Email : o3omoomin@gmail.com

LinkedIn : https://www.linkedin.com/in/devgoomin/

Instagram : https://www.instagram.com/9mean2/

보유 스킬 , tech skills

권구민 지원자는 다양한 포지션에서 실무 역량을 쌓은 지원자 입니다.

Frontend, Backend, LLM 포지션에서 여러 실무 경험을 쌓은 주니어 개발자 입니다.

Frontend skill은 아래와 같습니다.

- **React**
- **Javascript**
- **Redux(Thunk, Toolkit)**
- **Styled-Component**
- **Storybook**
- **Remix**
- **Typescript**
- **React Query,**

- **Tailwind-CSS**
- **Radix-Ui**

Backend skill은 아래와 같습니다.

- **Express**
- **PostgreSQL**
- **Prisma**
- **Kafka.js**

LLM skill은 아래와 같습니다.

- **LangChain**
- **Python**
- **llama-index**
- **Conda**

저는 이런 환경에서 협업에 능숙합니다.

- **Github**
- **VSCode**
- **ESLint**
- **GitLab**
- **Jupyter Lab**
- **Infisical**
- **MacOS**
- **Ubuntu**

현재 직장 및 경력 그리고 주요 성과

CN.AI(씨앤에이아이)

2023년 10월 ~ 재직 중

팀 : AI 기술 연구소 소속 Frontend Developer

주요 성과는 아래와 같습니다.

- **이미지 생성형 AI 플랫폼 Tivv 오픈 베타 출시**
- **CES 2024에서 Tivv 및 NexLook 프로젝트를 성공적으로 출품하였습니다.**
- **전문 문서 번역 영문화 (LLM Engineering)** 파이프라인을 구현하여 영문화 R&D과제 선정에 성공하였습니다.
- **다양한 프로젝트를 수행하면서 Backend 및 LLM Engineering 분야에서의 경험을 쌓았습니다.**
- **현재는 서울디지털재단_맥락정보 프로젝트를 진행하면서 NexLook 시제품 개발과 자사 플랫폼에 들어 갈 RAG ChatBot, 그리고 LLM BackOffice, 영문화 과제를 구현 중입니다.**

신도리코

2022년 01월 ~ 2022년 10월

팀 : 기획실 총무 팀

비 개발 직군으로 근무하였습니다.

자세한 경험 및 프로젝트에 대해 알고싶으시다면 “권구민 지원자의 프로젝트 및 경험에 대해 알려줘” 라고 입력해주세요, 혹은 해당 노션에서 확인하실 수 있습니다. https://fuschia-humor-318.notion.site/baeafb8b2d1f49dea525dab32db51b08?pvs=74

권구민의 보유 증명서

Coursera with DeepLearning.AI에서 수여한 Generative AI with Large Language Models 라는 증명서가 존재합니다.

해당 증명서는 아래 링크에서 확인 하실 수 있습니다.

https://www.coursera.org/account/accomplishments/verify/MPERW72KLYXU

권구민의 이력서 및 포트폴리오

권구민 지원자의 이력서 및 포트폴리오는 아래의 노션에서 확인하실 수 있습니다.

https://fuschia-humor-318.notion.site/baeafb8b2d1f49dea525dab32db51b08?pvs=4

권구민의 프로젝트 및 경험에 대해

저는 Frontend Position으로 근무 하면서 매우 다양 한 Task를 성공적으로 해결 해 나갔습니다.

LLM Engineering 프로젝트

프로젝트 : 전문 문서 번역 영문화 (LLM Engineering)

간단 설명 : 국가건설기준 용어와 파일의 형식을 유지하면서 영어로 번역하는 프로젝트 파이프라인 구축

스택 : **Python, Selenium, Beautifulsoup, Langchain, Claude3, ChromaDB, Conda**

주요 경험은 아래와 같습니다.

- **HWP Document Preprocessing and Conversion** : Selenium을 활용하여 전문 용어 및 HWP 파일을 스크래핑하고 HTML로 변환했습니다.
- **Korean Text Extraction and Processing** : BeautifulSoup4를 사용하여 HTML의 구조를 파악하고 번역에 필요한 한국어 텍스트를 추출하고 Chunking 작업을 수행했습니다.
- **Specialized Terminology Embedding** : 전문 용어 사전을 Chroma Vector Store에 로드하고 Embedding하여 사용했습니다.
- **System Prompting** : OpenAI의 Large Language Model (LLM)을 활용하여 시스템 프롬프팅을 진행하여 LLM의 답변을 유도했습니다.
- **LLM Translation Requests** : Semantic Search 기법을 사용하여 ChromaDB에 Query
- **LLM Response Handling and Translation Application**: : LLM의 응답을 수신하여 원본 HTML을 순회하며 한국어 텍스트를 번역문으로 적용했습니다.

- **현재는 영문화 R&D 과제를 착수하여 진행 중에 있습니다.**
  - **Anthropic 모델을 사용 하여 Context Issue 극복**
  - **RAG Task에 최적화 된 llamaindex를 도입하여 Inmemory Vector 구축**
  - **Prompt Engineering(FewShot)을 통한 Claude3 모델의 Response유도 및 한정 된 모델의 MaxToken의 효율적인 관리, 또한 전문 용어를 번역에 반영 할 수 있도록 Prompt engineering을 하였습니다.**
    - **Claude3의 200k Context로 인해 기존 파이프라인 Chunking 이슈를 해결 할 수 있었습니다.**
  - **원본 문서의 디자인과 포맷팅을 유지하기 위해 hwp → html 변환 시 번역이 필요 한 텍스트만 정제하여 각 태그에 고유값을 부여하여 성공적인 영문화 문서를 생성할 수 있었습니다.**
  - **모든 영문화 Task를 자동화하도록 FastApi및 서버 구축**

RAG Chat Bot (LLM Engineering)

간단 설명 : Retrieval Augmented Generation Chatbot를 구현 하여, 자연어 생성과 검색을 통합하여 사용자에게 정확하고 유용한 정보를 제공하는 프로젝트

스택 : **Python, llama-index, OpenAI, Pandas, Conda**

주요 경험은 아래와 같습니다.

- **RAG Research and Engineering 전담 구현**
- **Chunking and Vector Store** : 데이터로 활용 될 문서를 여러 크기의 청크로 분할하여, 각 Chunk에 대한 Vector Index Store를 구축 합니다. Retrieval을 구현 후 Query를 보낼 때 여러 크기의 청크를 참고하여, 문맥을 고려할 수 있도록 구현하였습니다.
- **Ensemble Retrieval** : 1개 이상의 Retrieval 를 Ensemble 하도록 구현 하였으며, 서로 다른 RAG Retrieval 강점을 활용함으로써 Ensemble Retrieval는 어떠한 단일 알고리즘보다 더 나은 성능을 얻을 수 있었습니다.
- **RecursiveRetriever** : 다양한 Index에서 Query를 통해 검색 된 결과를 종합하여, 이를 Recursive하게 검색한 후 최종 결과를 Response 하도록 구현하였습니다.
- **Rerank** : 최종 결과의 Response를 바탕으로 Rerank 알고리즘을 구현 하여 최종 결과의 Response Quality를 높이도록 구현 하였으며, 이는 더 나은 Response를 생성

하는데 더 나은 성능을 얻을 수 있었습니다.

- **Sentence Window Retrieval :** 기존 RAG 시스템의 텍스트 청크 크기 및 활용을 개선하여 문서를 더 작고 명확한 문장 수준으로 분할하고 특정 문장의 문맥을 고려하여 검색을 수행하도록 구현했습니다. 이를 통해 문맥을 더 잘 고려할 수 있으며, 문장 단위의 임베딩을 수행하여 검색 결과의 일관성과 유사성을 높일 수 있었습니다.
- **Mean Reciprocal Rank** : Response의 상대적 중요성을 분석하기 위해 메트릭 평가를 구현하였습니다. MRR을 사용하여, 각 Chunk Size에 대한 평가를 수행하였으며, 구현 한 Retrieval와 Baseline Retrieval 간의 성능을 비교하기 위해 다양한 평가 메트릭 사용하여 평가 하였습니다.
- **Eval Data Set Generator** : LLM의 성능을 평가하기 위해 Eval BenchMark Data Set의 Query와Response를 생성 하도록 평가 데이터를 구축하였습니다.
- **Evaluation** : Baseline Retriever와 비교하여 구현 한 Retriever가 더 좋은 성능을 보이는지, 정확성, 의미 유사성, 충실도 등의 평가 항목을 통해 두 Retriever간 성능을 정량적으로 측정하고 비교하도록 구현하였습니다.

FRONTEND 및 FULLSTACK 프로젝트

프로젝트 : Tivv

간단 설명 : 사용자가 입력한 프롬프트에 따라서 이미지를 자동으로 생성하거나 특정 이미지를 여러 스타일로 자유롭게 편집하는 생성형AI 플랫폼

스택 : **Remix(SSR), Typescript, IndexdDB, Tailwind, React window, Radix-UI, StoryBook, Express js, SSE, nodeMailer, Kafka, MinIO, Prisma(PostgreSQL), ESLint, Vitest**

주요 경험은 아래와 같습니다.

- **Tivv 오픈베타 서비스 출시 및 CES 2024 출품**
- **Prisma를 통한 ORM 시스템 관리**
- **Infisical을 통한 환경변수 관리**
- **Inference Enhancement Kafka** :Inference 과정의 성능 향상을 위해 Kafka를 도입하여 메시지 큐 시스템을 구축하였습니다. Broker 설정 후 producer 객체를 생성하여 Broker의 Topic을지정하여 Txt2img, Img2img, Variation Inference POST를 구현하였습니다.

- **SSE** : Inference Img 결과를 받기 위해 Remix에서 SSE 엔드포인트를 구축 후, Kafka Topic으로 전송 된 Request가 Inference가 끝난 후 SSE의 엔드포인트로 알림을 보내면 MinIO에 Access하여 Img를 렌더링하도록 구현하였습니다.
- **Payment** : TossPayment SDK를 통한 Tivv 플랫폼 내 Coin 구매 시스템을 구현하였습니다. (일반 결제)
- **Search History** : 별도의 검색기록 데이터 저장소를 구현하기 위해 IndexdDB를 사용하여 검색기록을 구현하였습니다.
- **Coin Count :** txt2Img, Img2img, Variation을 통한 Request가 일어나고, 성공적으로 Inference가 됐을 때 사용자의 Coin이 차감되는 시스템을 구현하였습니다.
- **Send Email** : NodeMailer를 활용하여 이메일 발송을 관리하였습니다. 사용자의 보안을 강화하기 위해 이미 가입된 사용자 또는 탈퇴한 사용자에게도 이메일을 발송하는 기능을 구현하였습니다.
- **MinIo** : 각 사용자마다 고유한 엑세스 키와 시크릿 키를 이용하여 Tivv의 Inference Img 등을 관리하였습니다.
- **Server Cookies :** 스테이징과 프로덕션 서버 간 로그인 시 쿠키 공유가 발생할 수 있는 에러를 방지하기 위해 각 서버에 고유한 쿠키 이름을 부여하여 관리하였습니다.
- **Storybook** : Storybook을 사용하여 공통 팝업과 사용자 프로필 UI, Global Navigation Bar를 독립적으로 개발하고, 드롭다운 기능 등을 포함한 각 컴포넌트의 상호작용을 효과적으로 확인하며 개발하였습니다.
- **Markdown Reader** : 이용약관과 개인정보 처리방침 팝업을 구현하기 위해 Markdown 형식을 HTML 형식으로 렌더링 하도록 구현하였습니다.

프로젝트 : NexLook

간단 설명 : 맥락정보 기반 CCTV로, 실시간 사건을 포착하여 분석하고, 맥락을 파악하여 범죄 예방에 기여하는 차세대 인공지능 CCTV

- **CES 2024 출품**
- **서울디지털재단_맥락정보 프로젝트 진행 중**
- **NexLook Frontend, Backend 구현**
- **Codec Incoding** : 브라우저에서 Decode 할 수 있도록 FFmpeg을 사용 하여 RTSP 코덱을 MPEG1으로 인코딩하고, 이를 WebSocket으로 송출 하도록 구현하였습니다.
- **CCTV Streaming :** jsmpeg을 사용하여 Express WebSocket으로부터 수신된 MPEG 비디오 데이터를 디코딩하여, 캔버스에 렌더링되도록 구현하였습니다.

- **Multi View Streaming** : 사용자가 선택한 RTSP 스트리밍을 동적으로 변경할 수 있도록 Redux를 사용하여 State를 관리하고, 선택된 RTSP 스트리밍이 변경될 때마다 해당 스트리밍에 대한 새로운 RTSP를 로드하도록 구현하였습니다.
- **Change Screen Mode** : 사용자가 화면을 전체 화면 또는 분할 화면 모드로 전환할 수 있도록 구현하였습니다. 이를 통해 사용자는 필요에 따라 CCTV 스트리밍을 최적화된 방식으로 볼 수 있습니다.
- **RESTful API를 통한 DeepStream 요청 처리** : DeepStream(AI Inference)으로부터 위험 요소가 발견 되면, Express에서 요청을 받아들일 수 있도록 RESTful API를 구현하고, 이를 처리하는 핸들러를 구현하였습니다.
- **Img2txt** : RESTful API를 통해 수신한 데이터를 정제하고, base64 이미지를 인코딩하여 OpenAI Vision LLM을 호출하여 img2txt를 구현하였습니다.
- **Server Send Event** : 클라이언트와 서버 간의 실시간 통신을 위한 SSE 엔드포인트를 구현 후 Img2txt의 생성 된 응답과 RESTful API 데이터를 조합하여 데이터 객체를 클라이언트로 실시간 전송되는 알림을 구현하였습니다.
- **Threat History :** 선택한 스트리밍에 대한 위험 탐지 기록을 동적으로 렌더링하도록 구현했습니다. 이를 위해 SSE를 통해 전송된 base64 데이터를 디코딩하여 렌더링하였습니다.
- **SSE Notification** : SSE 엔드포인트로부터 수신한 데이터를 RTSP 스트리밍 Canvas에 동적으로 렌더링하고, 맥락 정보와 함께 제공하기 위해 구현하였습니다. 또한, 다양한 상황에 따른 맥락 정보를 렌더링할 수 있도록 재사용 가능한 컴포넌트를 개발하였습니다.

LLM 분야에 어떠한 지식과 경험이 있는지

프로젝트에 대한 설명이 아닙니다. 아래 내용은 프로젝트로 진행 한 것이 아닌 권구민 지원자가 공부하면서 경험 한 것들입니다.

현재 회사에서 Generative AI 관련 직무에 대한 경험을 쌓으면서 LLM 관련 지식에 흥미가 생기기 시작했습니다.

그래서 Coursera with DeepLearning.AI에서 지원하는 Generative AI with Large Language Models의 교육을 들으며, ML 지식과 관련 된 Generative AI LLM의 Life

Cycle과 Transformers, Fine-tuning, RLHF 등 실무 역량에 필요 한 경험을 할 수 있었습니다.

저는 아래와 같은 경험과 지식이 있습니다!

- Transformers Architectrue : Transformers Architectrue의 종류, Flow, 학습 방법과 Transformers의 전체적인 프로세스가 어떻게 동작하는지에 대한 지식이 있습니다.
- Prompt Engineering 및 ICL 학습: Prompt Engineering을 통해 모델의 리소스 한계를 극복하고, In-Context Learning을 효과적으로 적용하는 방법을 경험했습니다.
- Pre-training LLM: BERT, GPT 등 다양한 LLM 모델의 사전 학습 방법을 이해하고 있습니다.
- Fine-tuning: BaseModel을 특정 작업에 맞게 학습 시킨 경험이 있습니다.
- Single task 및 Catastrophic forgetting 극복: Single task를 극복 하는 방법과 Catastrophic forgetting을 해결 하는 방법 및 경험이 있습니다.
- Multitask-Instruction fine-tuning: 다중 작업에 대한 모델 튜닝 방법에 대한 지식과 경험을 갖고 있습니다.
- Fine-tuning된 모델 Evaluation (ROUGE, BLEU) 및 BanchMark측정을 통해 비교 및 평가에 대한 경험
- PEFT LoRA를 활용한 Fine-tuning: LoRA 방식의 Fine-tunung을 구현하여 텍스트 요약 작업을 효율적으로 수행한 경험이 있습니다.
- RLHF를 통한 Human Align 모델로 학습시킨 경험이 있으며, Reward 모델과 RL알고리즘을 통해 LLM을 Human Align 모델로 학습시킨 경험이 있으며, Reward hacking을 방지를 위해 KL을 사용하여 RLHF 프로세스를 진행 하였습니다.
- Fine-tuning된 모델 Evaluation: ROUGE, BLEU 등의 지표를 활용하여 모델 평가와 성능 비교를 수행한 경험이 있습니다.
- 현재 재직중인 회사에서 RAG와 관련 된 프로젝트를 진행중에 있습니다.
