김도현

00dh.kim@gmail.com | +82 10 7730 8994 | linkedin.com/in/00dhkim | github.com/00dhkim

AI Engineer

Python, PyTorch, RAG, LangGraph, LLM Evaluation, Context Engineering, AI Observability

Vision

측정과 증명(show and prove)을 통해 AI의 가치를 실현하는 엔지니어입니다.

저는 '측정 가능한 비효율'에서 '좋은 문제'가 발견된다고 믿습니다. AI 도입의 가치는 감이 아닌 데이터로 증명해야 하며, 모든 의사결정은 숫자로 뒷받침되어야 합니다.

문제를 표면에서 해결하는 데 그치지 않고, 근본적인 원인을 찾아 문제의 본질을 파고듭니다. 평가 프레임워크를 구축하고, LLM 파이프라인의 퀄리티를 높이며 사용자의 체감 효용을 높입니다.

기술 편향적 사고방식을 넘어, 사용자의 불편함을 제거하고 비즈니스 가치를 창출하는 '진짜 임팩트'를 만들고자 노력합니다.

Employment

국방과학연구소 | 현역 연구원(과학기술전문사관) | 2023.06. - 현재

전사 최초의 AI Assistant '초석이' 개발

LangChain, LangGraph, Ragas, Phoenix, Langfuse, vector DB, BM25, vLLM

- 전사 최초 AI Assistant의 **'규정 검색 챗봇' 기능을 기획·구현**. 사용자 설문과 인터뷰를 통해 임팩트 지점을 설정하고 유저테스트를 통해 검증, 사용자 체감 시간을 62초에서 8초로 87% 효율화 성공.
- Ragas 오픈소스로 **RAG pipeline 평가 프레임워크를 구축**하여 데이터기반 의사결정을 실현. Ragas에서 생성한 test set의 invalid data를 발견, context 최적화로 valid 비율을 88% -> 97%로 개선.
- **retriever의 반복적 개선**으로 72.2%였던 context recall을 95%까지 개선. multi index, BM25, reranker 전략을 고려하고 Ragas와 Phoenix 오픈소스 도구로 정량적 수치를 평가해 사용자 경험 개선.
- **경량 로컬 모델을 위한 table parser** 개발. 행/열을 동시에 참조하는 고난도 표 인식에 실패하는 문제를 식별하고 serialized parsing 기법으로 해결, 사내 table benchmark 샘플로 검증 성공.
- **AI Governance를 총괄**하여 방첩사령부의 100+가지 보안 규제를 만족하는 아키텍처 설계 및 구현. 사내 최초의 LLM guideline을 제정하여 C-level 승인 획득 및 10억원 과제비 확보.

멀티에이전트 강화학습 기반 임무 수행 의사결정 시스템 개발

Pytorch, Reinforcement Learning, Transfer Learning, Multi-Agent System

- 미래 전장 상황에서 다수/이종 로봇들에 대해 임무를 효율적으로 수행하기 위한 임무급 의사결정 Multi-Agent 시스템 개발.
- Agent 학습이 수렴하지 않는 문제를 해결하기 위해, Agent의 입력에 해당하는 state space를 normalize하여 일관된 feature 정보를 획득하도록 개선.
- 모델의 robustness를 향상시키기 위한 Multi-Agent Network Randomization 기법 제안, smac benchmark에서 임무 성공률에 대해 기존 82% 대비 90%로 향상.

Publications / Patents

- Reinforcement Learning-based Fault-Tolerant Control for Quadrotor with Online Transformer Adaptation, 1저자, ICRA 2025 Workshop. [link]
- Multi-Agent Network Randomization Method for Robust Knowledge Transfer in Deep Multi-Agent Reinforcement Learning, 1저자, ICCAS 2024. [link]
- Autonomous Collaboration Control for Manned-Unmanned Complex Systems and Its Compositions, 2저자, 군사과학기술학회지, 2024, Best Paper Award. [link]
- Stochastic Initial States Randomization Method for Robust Knowledge Transfer in Multi-Agent Reinforcement Learning, 1저자, 군사과학기술학회지, 2024. [link]
- [특허] 에이전트 제어 방법 및 그 방법을 수행하는 전자 장치, 2024. [link]
- [특허] 다중 에이전트를 위한 강화 학습 방법 및 이를 위한 전자 장치, 2025. [link]

Experience

대구를 빛내는 SW해커톤 주최 | TF팀장 | 2022.06. - 2022.10.

- 학생들이 개발 성과를 공개적으로 전달할 기회가 부족하다는 문제 발견, 프로젝트 기반 성장과 상호 동기부여를 통해 성취를 경험할 수 있도록 해커톤 기획·운영을 총괄.
- 대구디지털혁신진흥원 및 스타트업과 협력하여 총 2,200만원의 예산 확보, 221명 참가자 등록.
- 행사 후 익명 설문 결과, '첫 프로젝트를 통해 자신감을 얻었다', '부족한 부분을 깨닫고 자극이 되어 공부 방향을 깨달았다'는 응답과 함께 성공적인 해커톤으로 마무리.

경북대학교 컴퓨터학부 학생회장 | 2022.01. - 2022.12.

- 1,000+명의 학생을 대표하여 50명의 운영진과 함께 학생회를 운영, 복지프로그램 및 학생활동을 총괄하여 3,000만원의 예산 집행.
- 6개의 외부기관과 소통하며 공식적/비공식적 커뮤니케이션으로 협상을 주도. 기관 간 의사결정 교착 상태(Deadlock)을 능숙하게 대처.
- 실수한 점과 배운 점을 인수인계 자료에 솔직하게 기록하여 업무연속성 확보에 기여.

Education

경북대학교 컴퓨터학부 졸업 | 2019.03. - 2023.02.

- GPA 4.24 / 4.5 (156학점)
- 정보보안연구회 회장(2020.03. - 2021.07.) 공공기관 웹취약점 진단 사업 수행
- 학부연구생(2020.07. - 2020.12.) Clang Static Analyzer에 대한 CWE 벤치마킹 수행

Awards

대한민국 인재상(2022) [매일신문]
소프트웨어 역량검정 성적우수자(2022) [과기부]
과학기술전문사관 밀리테크 챌린지(2021)
K-전차 인공지능 챌린지(2022)
한국남부발전 웹서비스 정보보안 경진대회(2021)

1. 연구소 최초 AI Assistant 구축을 통한 행정 업무 생산성 혁신

Situation

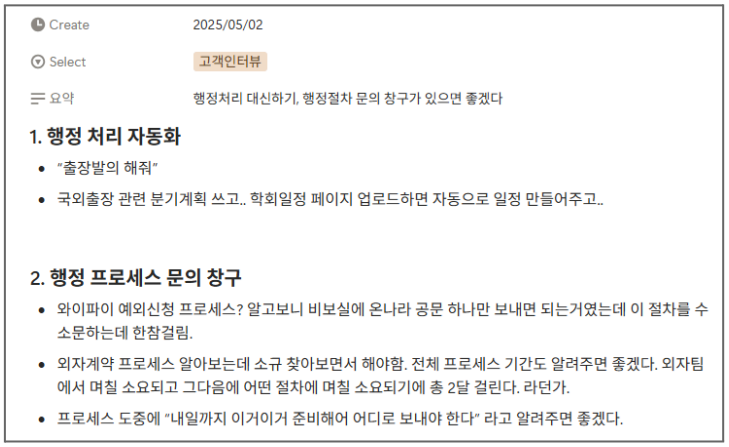
국방과학연구소는 가급 보안 시설로서, 외부 인터넷이 차단된 환경에서 복잡한 보안 규제와 반복적인 행정 업무가 산재해 있습니다. 가령, 업무용 사진 촬영을 위해서 '장비반입허가 - 촬영허가 - 저장매체사용허가'의 단계적 승인 절차가 필요하며, 모든 승인까지 하루 이상이 소요됩니다. 이러한 **단순 반복적 행정 업무에 쏟는 시간이 일평균 2.53시간**에 달하며, 설문조사 참여자 **781명 중 88.4%가 행정 업무에서 비효율을 체감**하고 있다고 응답했습니다.

Task

저는 이러한 문제를 AI로 개선하고자 했으나, open-weight LLM의 보안 위협을 관리하기 어렵다는 이유로 보안 부서의 반대를 마주했습니다. 그러나 평소 AI를 활용한 행정 효율화의 필요성을 적극적으로 피력해온 점을 인정받아, 부소장 직속 '전사적 AI 혁신 TF'에 현역연구원으로는 유일하게 참여하게 되었습니다. 저는 2달간 3천 명의 구성원을 위한 AI 솔루션을 기획부터 구현, 배포하는 업무를 맡았으며, 이를 통해 AI 응용의 best practice를 제시하여 정책 부서를 설득하고 전사적 AI 도입의 환경을 마련하는 책임을 부여받았습니다.

Action

전직원 설문조사와 심층 인터뷰를 통해, '복잡한 행정 절차를 파악하는 과정'이 업무 효율을 저해하는 가장 큰 원인임을 식별했습니다. 따라서 저는 2달 내 구현가능한 솔루션으로, 내부 규정 문서에 대해 질의응답하는 '규정 검색 AI Assistant'를 기획했습니다. 저는 이를 구현하기 위해 문서파싱, vector DB 구축, RAG 파이프라인 개발을 담당했습니다. (개발 과정에서의 기술적 challenge 및 해결책은 다음 페이지에 이어서 설명드릴 예정입니다.)



Create 2025/05/02

Select 고객인터뷰

요약 행정처리 대신하기, 행정절차 문의 창구가 있으면 좋겠다

1. 행정 처리 자동화

- "출장발의 해줘"
- 국외출장 관련 분기계획 쓰고.. 학회일정 페이지 업로드하면 자동으로 일정 만들어주고..

2. 행정 프로세스 문의 창구

- 와이파이 예외신청 프로세스? 알고보니 비보실에 온나라 공문 하나만 보내면 되는거였는데 이 절차를 수소문하는데 한참걸림.
- 외자계약 프로세스 알아보는데 소규 찾아보면서 해야함. 전체 프로세스 기간도 알려주면 좋겠다. 외자팀에서 며칠 소요되고 그다음에 어떤 절차에 며칠 소요되기에 총 2달 걸린다. 라던가.
- 프로세스 도중에 "내일까지 이거이거 준비해어 어디로 보내야 한다" 라고 알려주면 좋겠다.

심층 인터뷰 기록

Result

파일럿 프로젝트의 결과물로 AI Assistant '초석이'를 성공적으로 배포하였으며, 수십 페이지가 넘는 각종 규정을 일일이 검색하며 자료를 찾던 사용자의 부정적 경험을 해소하여 자연어 질의만으로 사용자가 원하는 정보를 찾게 개선했습니다. 베타테스트에서 기존 시스템을 통해 **62초 소요되던 사용자 체감 시간을 8초로 단축, 소요시간이 87% 개선된 결과**를 보였습니다. 이러한 성과는 생성형 AI 도입 과정에서 C-level에서 제기된 보안 이슈를 해소하였으며, 10억원 규모의 과제비를 할당받는 결과로 이어졌습니다.

2. 데이터기반 의사결정을 위한 RAG 평가 프레임워크 구축 및 개선

Situation

LLM은 강력하지만, 민감한 기술이라고 생각합니다. 어떤 양자화 모델을 채택할지, 어떤 context를 전달하는지에 따라 성능이 크게 달라질 수 있기 때문입니다. 따라서 이러한 변수에 따른 **성능 개선을 객관적으로 수치화**하는 평가 프레임워크를 구축하여, **데이터 기반 의사결정**을 가능하게 해야 한다고 생각했습니다.

Task

'객관적인 평가 프레임워크 구축'을 위해 다음의 옵션들을 고려했습니다.

- 사용자 데이터를 기반으로 Golden Set을 구축하는 방법은 가장 객관적으로 RAG 파이프라인을 평가하는 방법입니다. 하지만 비용이 발생하며, 배포 전 평가 단계에서 활용하기 어렵다는 단점이 있습니다.
- Ragas 오픈소스 도구는 수정이 용이하고 자체적으로 QA set을 생성하는 TestsetGenerator 기능이 존재합니다. 이를 이용해 Silver Set을 구축하는 것이 비용효율적일 것이라 판단했습니다.

Action

Ragas로 테스트셋을 생성하고 그 결과물을 스크리닝했을 때, 결과물의 퀄리티가 예상보다 부족한 문제를 인식했습니다. 특히 일부 샘플에서 "출장규정 1조는 무엇인가요?", "광역시장은 뭐에 관련된거야?" 등과 같이 사용자 시나리오에 부합하지 않거나 과도하게 쉬운 샘플이 포함되고 있었습니다.

이를 개선하기 위해 Ragas 내부 구조를 공식 문서와 레포지토리를 통해 공부하고, 결과적으로 Ragas가 내부적으로 문서들을 knowledge graph로 변환한 뒤에 테스트셋을 생성한다는 것을 알았습니다. 그리고 이 변환 과정에서 쓰이는 entity extractor(고유어 추출기)가 단순 정보(1985.10.15., 제1조)나 일반명사를 출력하는 에러를 일으키고 있다는 것을 발견했습니다. entity extractor는 LLM을 통해 동작하는데, 프롬프트가 과도하게 일반적이고 사내 데이터에 맞지 않은 few-shot example을 담고 있었습니다. 저는 instruction에 제약조건을 추가해 오탐을 줄이고, self-check 메커니즘을 추가하였으며, 사내 데이터에 알맞는 한국어 few-shot prompting을 추가해 사용자 시나리오를 반영하도록 개선했습니다.

Result

이러한 개선 후에 200건이 넘는 규정 문서들을 대상으로 다시 테스트셋을 생성하였고, **88%였던 valid data 비율을 97%까지 향상**시킬 수 있었습니다. 이로써 **고품질 비용효율적 데이터셋을 통한 신뢰할 수 있는 의사결정에 기여**했습니다. 또한 이러한 개선점에 대한 ragas 공식 레포지토리에 issue를 작성하는 동시에, 사내 팀 주간 세미나에서 발표하였습니다. 해당 경험에서 배운 인사이트와 **llm-as-a-judge의 응용 사례를 공유**하여 팀 내 AI 활용을 촉진하였습니다.

3. Retriever 성능 개선을 위한 반복적 실험 - 분석 - 평가

Situation

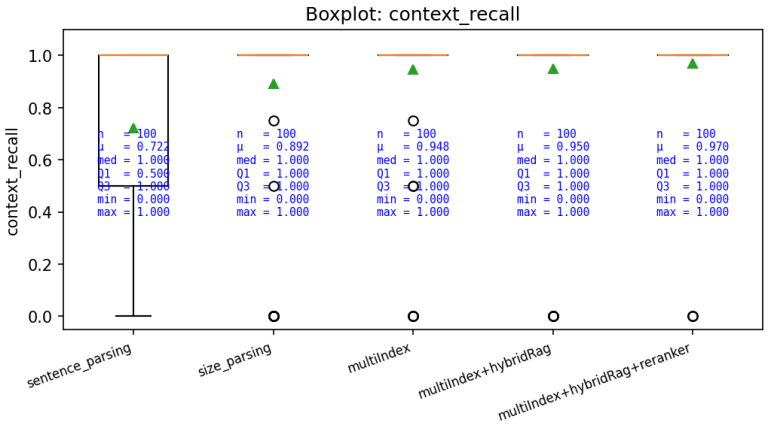
구축한 AI Assistant '초석이'의 성능을 phoenix 오픈소스 도구로 모니터링하는 과정에서, **"관련된 문서를 찾을 수 없습니다."라는 출력**이 과도하게 자주 발생하는 문제를 확인했습니다. 사용자 질문에 관련된 문서가 vector DB에 포함되어 있음에도, 해당 chunk가 **retrieve 되지못해 사용자의 질의에 답변하지 못하는 상황**이었습니다.

Task

이 문제를 해결하기 위해, 가설을 설정하고 실험을 통해 검증하는 사이클을 반복하여 retriever의 context recall을 단계적으로 상승시키고자 노력했습니다.

Action

- **[Baseline]** 초기 retriever는 단일 인덱스 vector search로 구현되었으며, 당시 context recall는 72.2%였습니다.
- **[multi-index 전략]** 질의 의도에 따라 필요한 context의 '해상도'가 다름을 발견했습니다. 세부 사항에 관한 질문은 문장 단위 chunk로 파싱하여 retrieve 성공률을 높였고, 개괄적인 맥락과 관련된 질문은 문단 단위 chunk로 파싱하여 주위의 정보를 함께 가져오도록 설계했습니다. 두 파싱 방법에 따른 vector DB를 별도로 구축하는 multi-index 전략을 도입하여, context recall을 94.8%로 향상시켰습니다.
- **[Hybrid RAG 도입]** dense vector는 의미적 유사성 파악에 유리하지만, 임베딩 모델이 학습하지 못한 고유명사나 키워드 매칭에는 한계가 있음을 확인했습니다. 이를 상호 보완하는 BM25와 RRF(Reciprocal Rank Fusion) 알고리즘으로 결합하는 Hybrid RAG를 구축하여 context recall을 95%까지 개선했습니다.
- **[Reranker 검토]** bi-encoder가 획득하지 못하는 맥락적 유사성을 추출하기 위해 cross-encoder 기반의 reranker를 실험했습니다. context recall은 97%로 상승했으나, reranker 노드에서 P99 latency가 60초까지 증가했습니다. 실서비스에서 2%p의 recall 상승이 60초의 latency를 설득하지 못한다고 판단하여, 최종적으로 reranker를 제외한 RAG 파이프라인을 서비스에 적용했습니다.



retriever 최적화에 따른 context recall의 점진적 상승

Result

반복적 최적화를 통해 72.2%였던 context recall을 95%까지 향상시켜 '답변 불가' 문제를 해결했습니다. 특히 성능과 응답 속도 사이의 균형점을 찾아 적용함으로써, 서비스 안정성을 해치지 않고도 사용자로부터 "이젠 RAG가 필요한 문서를 정확히 찾아준다"는 평가를 받았습니다.

4. 경량 모델에서의 retrieve 성능 개선을 위한 serialized parsing 기법 도입

Situation

RAG 파이프라인의 성능을 평가하는 과정에서, **표의 본문이 context에 포함되었을 때 그 결과물이 만족스럽지 못한 문제**를 겪었습니다. 올바른 chunk가 적절히 retrieve 되더라도 할루시네이션이 나타나거나 잘못된 정보를 인용한다는 이슈를 확인했습니다.

Task

ChatGPT나 Claude 등의 외부 모델에 기존 방식의 prompt와 context 쌍을 입력할 경우, 기대하는 답변을 받을 수 있었습니다. 그러나 동일한 입력을 30B 이하의 경량 모델에 적용하자 성능 저하가 발생했습니다. 복잡한 표의 내용을 경량 모델이 인식하지 못한다는 가설을 세웠고, 이를 검증하기 위해 구조적으로 정보를 인지할 수 있는 개선된 표 파서 개발을 추진했습니다.

Action

- 우선 100여 개의 내부 문서를 눈으로 읽으며, 표의 스타일을 simple, relational, one-hot의 세 타입으로 분류했습니다. 표 스타일에 따른 별도의 파싱 방식을 고려했지만, 표 스타일 분류에 큰 업무량이 부하될 것이 우려되었습니다.
- 다음 대안으로 HWP 형태의 표를 markdown 및 html로 형식화하는 방법을 시도했습니다. 형식화된 포맷을 사용할 경우 colspan이나 rowspan를 표현할 수 있고, 또 이는 일반적으로 LLM이 잘 인식하는 형식으로 알려져 있습니다. 하지만 경량 모델의 한계로 인해, 길이가 길거나 구조가 복잡한 표에 대해서는 여전히 인식 실패 문제가 발생했습니다.
- 이 문제에 대해 ChatGPT와 토론하던 중, serialized parsing 기법을 발견했습니다. `휴가일수=직급; 선임급=15; 책임급=20;` 형태로 파싱하는 방법으로, 셀 각각의 값에 label을 붙여 LLM이 인식하기 쉬운 chunk를 만드는 방법이었습니다.

Result

serialized parser의 성능을 평가하기 위해, 내부 데이터를 바탕으로 사내 table benchmark를 새롭게 만들었습니다. 그리고 이를 serialized parser가 적용된 로컬 경량 모델로 평가했을 때, **모든 샘플에 대해 표 인식에 성공**했습니다. 단순히 RAG 구축에 그치지 않고, 반복적 시행착오로 **사용자가 체감하는 만족도를 높이기 위해** 문제를 집요하게 탐색하고 해결했습니다. 반복적 시행착오를 통해 완성도 있는 제품 개발을 경험했습니다.