Fast campus

the Red.

LLM 마스터클래스 #6

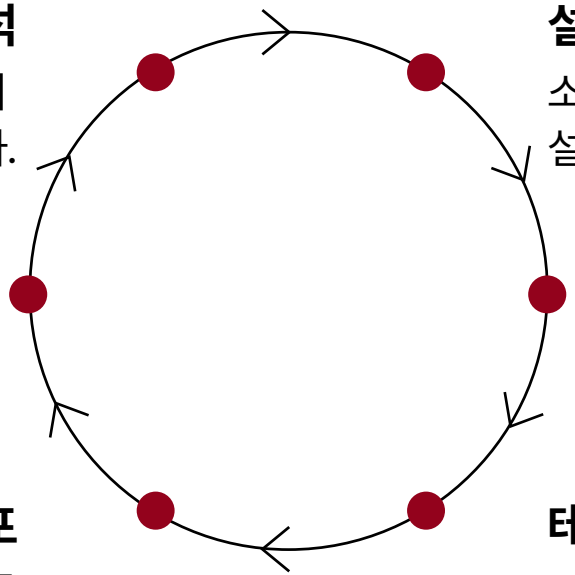
LLM 서비스 운영: 아키텍쳐와 도입

Part 6-1.
Software Life Cycle + Devops 역사, 설명

소프트웨어 라이프사이클 (초간단 요약)

요구사항 분석
어떤 소프트웨어가 필요한지
정확히 파악합니다.

설계
소프트웨어의 구조, 기능, 디자인 등을
설계합니다.

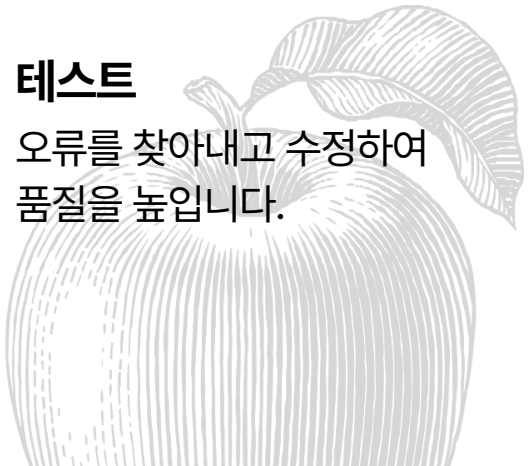


유지보수
지속적인 업데이트와 문제 해결을
통해 소프트웨어를 관리합니다.

개발
실제로 코드를 작성하여
소프트웨어를 만듭니다.

배포
사용자에게 소프트웨어를
제공합니다.

테스트
오류를 찾아내고 수정하여
품질을 높입니다.



DevOps 역사

원래 그런거 없었음

2000년대 초반

애자일 방법론 소프트웨어 개발에서 애자일 방법론이 널리 채택되면서, 개발과 운영 간의 협력이 필요해짐. 만들고 넘기는 것이 아님.

2008년

DevOps 용어의 등장
: 벨기에의 컨퍼런스에서 패트릭 드보아(Patrick Debois)가 DevOps라는 용어를 처음 사용

2009년

첫 DevOpsDays
: 벨기에에서 첫 DevOpsDays 행사

도구와 프레임워크의 발전

Jenkins, Docker, Kubernetes와 같은 도구가 등장. 지속적 통합(CI)와 지속적 배포(CD) 파이프라인이 널리 채택 (자동화된 빌드, 테스트, 배포 파이프라인을 구축하여 개발 주기를 단축하고 안정성을 높임)

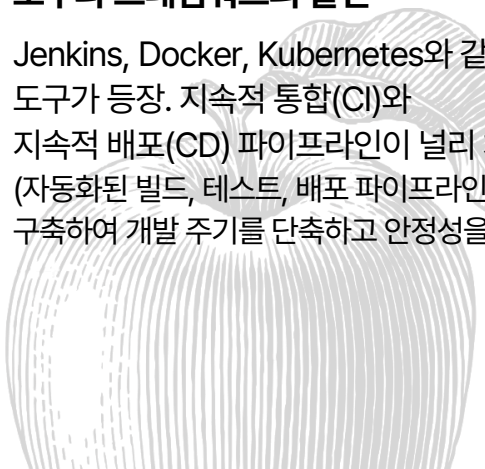
클라우드 컴퓨팅의 부상

클라우드 환경은 인프라 확장성과 유연성을제공하여 DevOps 도입을 가속화했습니다.

마이크로서비스 아키텍처 도입

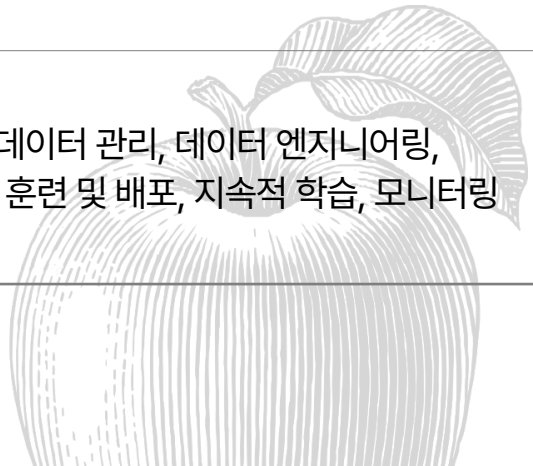
애플리케이션을 작은 서비스 단위로 분할하여 개발 및 배포 속도를 높이고 유연성을 확보했습니다.

구글에서 직함으로 만들기 시작



DevOps vs MLOps

	DevOps	MLOps
개념	소프트웨어 개발(Development)과 IT 운영(Operation)의 통합	머신 러닝(ML) 모델 개발과 운영(배포, 유지 보수)의 통합
목적	소프트웨어 개발과 배포를 자동화하고 효율적으로 관리	ML 모델의 개발, 자동화, 배포, 모니터링, 관리 자동화
주요 요소	자동화, 지속적 통합(CI), 지속적 배포(CD), 모니터링, 인프라 관리, 개발과 운영의 협업	데이터 관리, 데이터 엔지니어링, 모델 훈련 및 배포, 지속적 학습, 모니터링



MLOps 그리고 이제는 LLMOps

가장 핫 했던 분야

MLOps topped LinkedIn's Emerging Jobs ranking,
with a recorded growth of 9.8 times in five years.

“이제 LLMOps”

Part 6-2.
LLMOps stack

LLM Ops Stack

LLM Ops 스택이란 무엇인가요?

대규모 언어 모델(LLM)을 실제 환경에서
원활하고 효율적으로 작동시키는 도구 및 프로세스.

왜 중요한가요?

LLM은 강력하지만 신중한 관리가 필요.
스택은 기업이 LLM을 최대한 활용하면서 위험을 최소화.

- 효율적인 자원 관리
  : 컴퓨팅은 비쌈. 추론도 비쌈. 훈련도 비쌈.
- 모니터링 및 유지보수
- 데이터 관리
- 확장성
- 보안 및 규정 준수
- 사용자 피드백

LLM Ops Stack

주요 구성 요소

- 데이터 파이프라인 (LLM의 원료) - 데이터 수집 및 전처리
- 모델 훈련 (LLM 가르치기) 학습 인프라 및 프레임워크, 하이퍼파라미터 튜닝 및 최적화
- 배포 (LLM을 실제로 사용)
  - containerization, orchestration
  - Docker, Kubernetes
  - 확장성 / 로드 밸런싱
- 모니터링 (성능 주시)
  - latency, requests completed
  - model drift
  - updates

LLM Ops Stack - 실제 문제들

데이터 ▶ 훈련 ▶ 서비스 만들기 ▶ 서비스 쓰기 ▶ 서비스 모니터링

LLM Ops Stack - 실제 문제들

- 프롬프트 버전 관리
- 배치/벌크 테스팅
- 데이터 파이프라인 인터페이스 만들고 관리하기
- 모델 스피드/가격/성능 요구에 따라 다른 모델 쓰기
- 모델 로드 밸런싱
- 병목현상 찾기
- 품질 저하 이유 찾기
- 데이터 retrieval 최적화
- Retrieved data 스토리지 최적화 및 관리
- 데이터 로깅 vs RAI 요구사항 맞추기
- Regression 테스트 만들기
- SLA, KPI 만들고 모니터링하기

Part 6-3.

LLM 스피드 / 가격 통제

LLM 속도 향상 전략

모델 경량화	• 가지치기 (Pruning): 불필요한 모델 파라미터 제거 • 양자화 (Quantization): 모델 파라미터의 정밀도 감소 • 지식 증류 (Knowledge Distillation): 작고 빠른 모델에 지식 전달
추론 최적화	• 배치 처리 (Batching): 여러 요청을 한 번에 처리 • 캐싱 (Caching): 자주 사용되는 결과 저장 및 재사용 • 하드웨어 가속 (Hardware Acceleration): GPU, TPU 등 활용
코드 및 알고리즘 최적화	• 효율적인 알고리즘 사용 • 병렬 처리 (Parallel Processing) • 비동기 처리 (Asynchronous Processing)

LLM 비용 절감 전략

클라우드 리소스 최적화	• 인스턴스 유형 선택: 필요에 맞는 적절한 인스턴스 사용 • 자동 확장 (Auto Scaling): 트래픽 변동에 따라 리소스 자동 조절 • 스팟 인스턴스 (Spot Instances) : 유휴 리소스 활용 (AWS, GCP, Azure, up to 90% off) • 컨테이너화: Docker, Kubernetes, 자원의 격리와 효율적 사용 • 서버리스 컴퓨팅
모델 압축 및 공유	• 모델 가중치 공유: 여러 모델 간 가중치 공유 • 모델 압축 (Model Compression): 모델 크기 축소
에너지 효율적인 하드웨어	• 저전력 GPU, TPU 사용 • 에너지 효율적인 데이터 센터 활용
모니터링 + throttling	

Part 6-4.
CI/CD 파이프라인

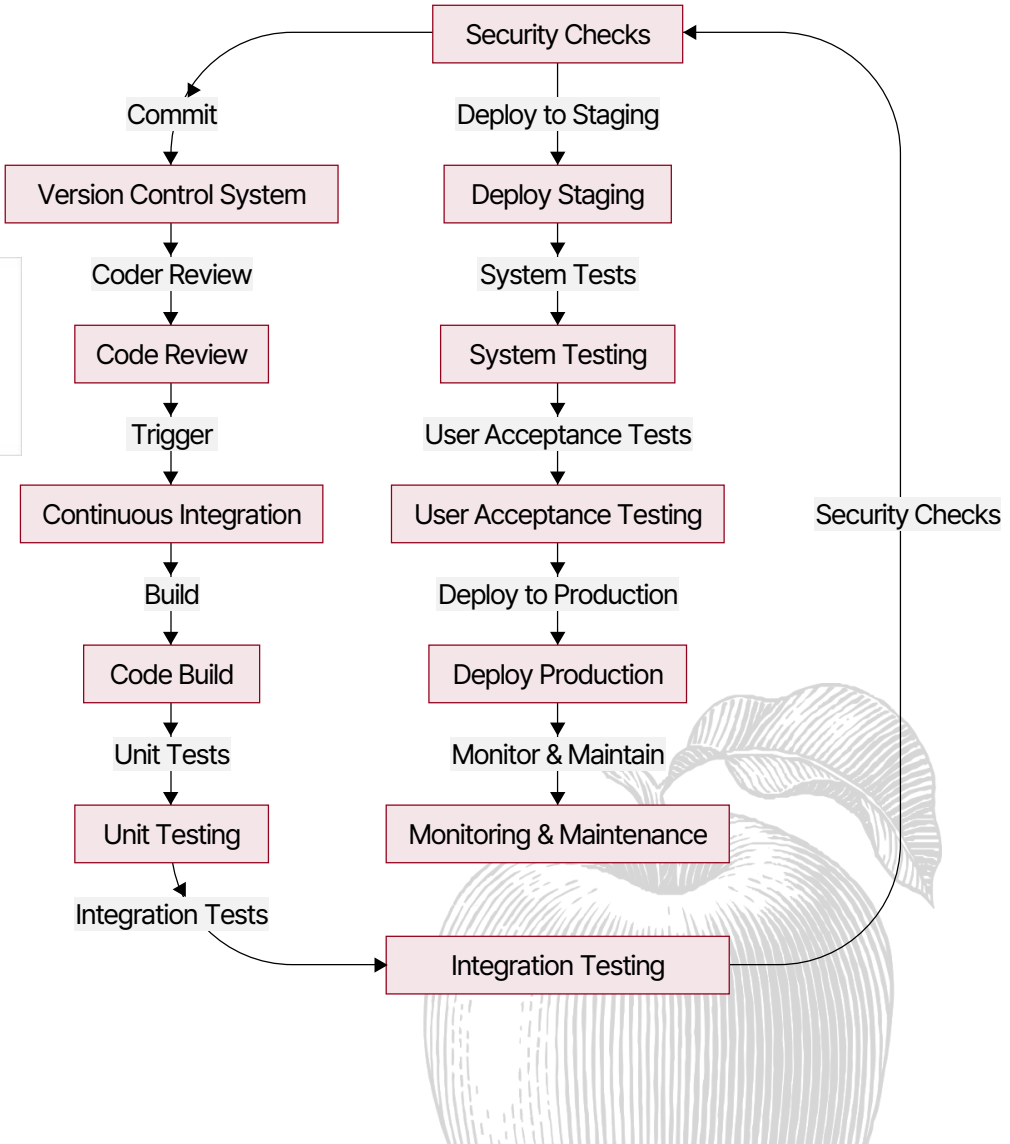
LLM 과 CI/CD

샘플 플로우 1

Navveen Balani on LinkedIn: #ai #devops #llm #cicd #automation #generativeai...

Multi-Modal Large Language Models (LLMs) are set to revolutionize the way we work. A key feature ...

linkedin

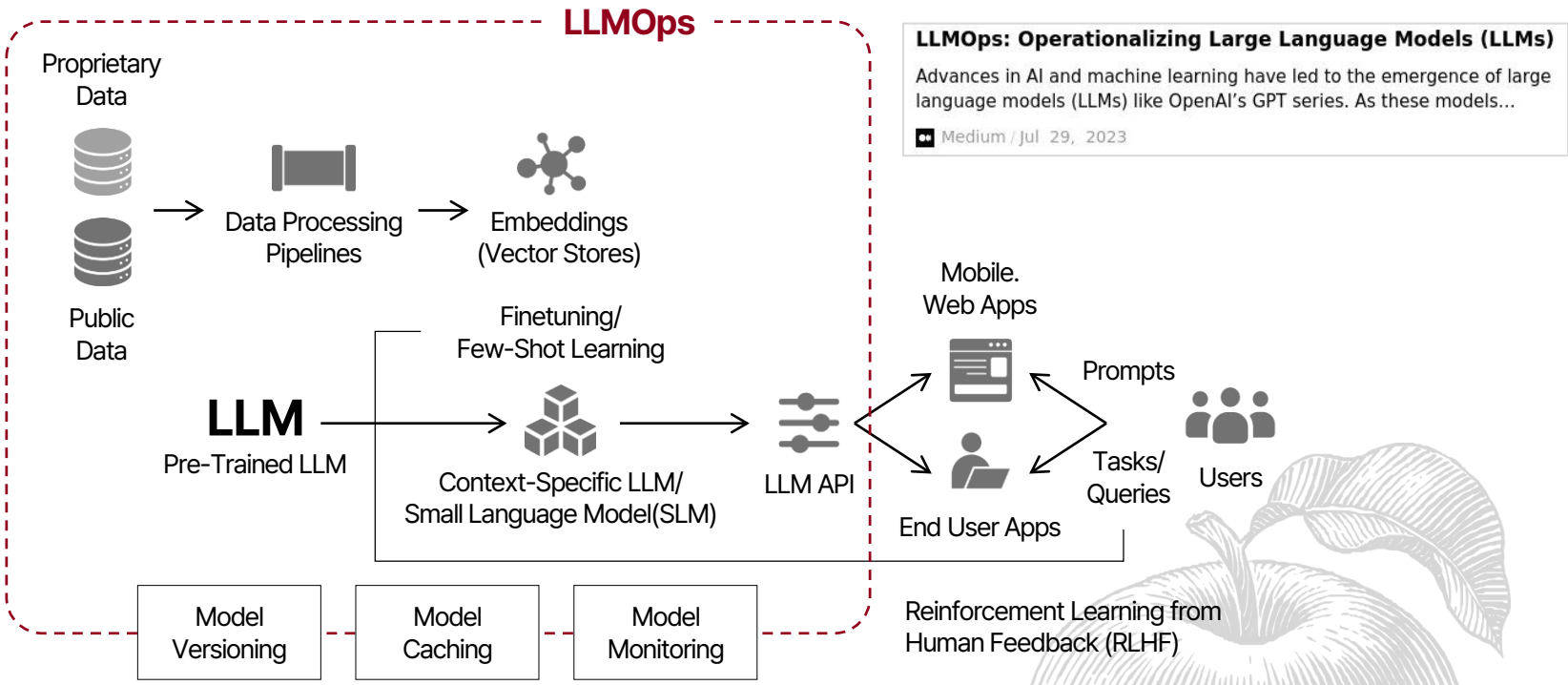


LLM 과 CI/CD

- 지속적 통합과 지속적 배포
- 자동 빌드 및 테스트 : 유닛 테스트, 통합 테스트, 모델 성능 테스트
- 버전 관리 - GIT for code, model version
- Docker/ Kubernetes 사용한 컨테이너 화
- 성능/지표 게이트
- EverGreen / canary 배포
- 롤백 전략
- “observability”
- criteria for going to prod
- collect user feedback

LLM 과 CI/CD

샘플 플로우 2



Part 6-5.
보안 문제 해결하기

LLM 과 보안

Prompts	• Prompt leakage • Prompt hijacking
Data	• Use RAG • Monitor "agents" : 아무거나 주문할 수 있게 비번 설정해두지 않기 • 로깅할 때 조심
세션 데이터 관리 - compliance 부터 체크	
재훈련할 때의 데이터 관리	
API call security	

이것을
guardrails
라고 부름

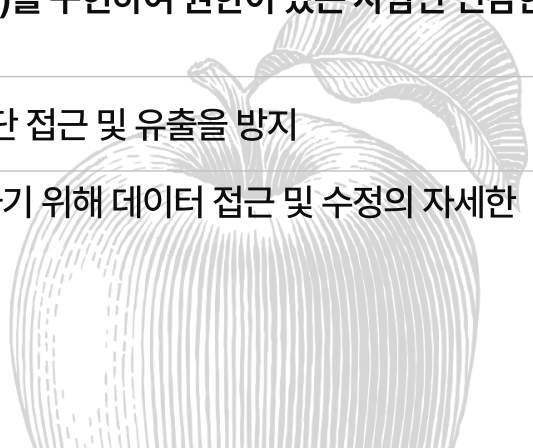
LLM 과 보안

1. 데이터 수집

데이터 프라이버시	**수집된 데이터가 데이터 보호 규정(GDPR, CCPA 등)을 준수하도록 함. 익명화**
데이터 유효성 검사	데이터 주입 공격을 방지하고 데이터가 악성 콘텐츠가 없도록 하기 위해 입력 데이터를 검증하고 정리
보안 전송	전송 중 데이터를 보호하기 위해 HTTPS, SSL/TLS 등의 보안 프로토콜을 사용

2. 데이터 관리

접근 제어	**엄격한 접근 제어 및 역할 기반 접근 관리(RBAC)를 구현하여 권한이 있는 사람만 민감한 데이터에 접근**
암호화	휴지 상태 및 전송 중인 데이터를 암호화하여 무단 접근 및 유출을 방지
감사 로그	무단 활동을 모니터링하고 법의학 분석을 지원하기 위해 데이터 접근 및 수정의 자세한 로그를 유지



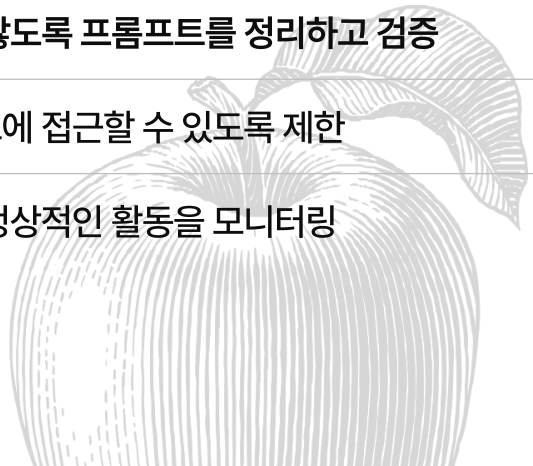
LLM 과 보안

3. 모델 학습

데이터 무결성	**데이터 중독 공격을 방지하여 모델의 행동을 조작하지 않도록 학습 데이터의 무결성을 보장**
프라이버시 보호 기술	학습 데이터의 민감한 정보를 보호하기 위해 차등 프라이버시와 같은 기술을 사용
모델 보안	학습 환경을 정기적으로 업데이트하고 패치하여 취약점을 방지

4. 프롬프트 관리

입력 정리	**주입 공격을 방지하고 악성 콘텐츠가 포함되지 않도록 프롬프트를 정리하고 검증**
접근 제한	권한이 있는 사용자만 프롬프트 관리 인터페이스에 접근할 수 있도록 제한
모니터링	시도된 공격을 나타낼 수 있는 의심스럽거나 비정상적인 활동을 모니터링



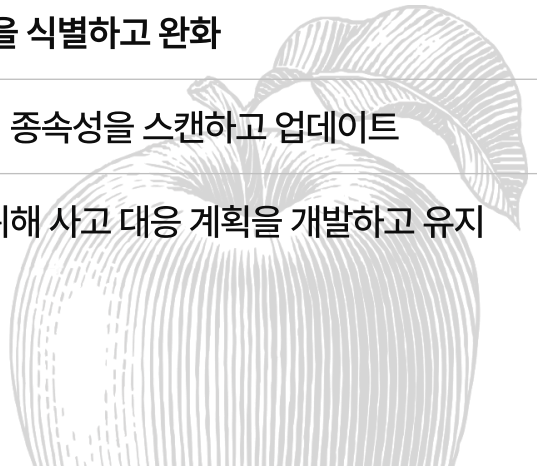
LLM 과 보안

5. 모델 배포

보안 환경	**모델을 잠재적 위협으로부터 격리하기 위해 가상 사설 클라우드(VPC)와 같은 보안 환경에 배포**
컨테이너 보안	컨테이너를 사용하는 경우 보안 구성 및 정기 업데이트를 통해 취약점을 완화
API 보안	인증 및 권한 부여 메커니즘, 속도 제한, 입력 유효성 검사로 API 엔드포인트를 보호

6. 애플리케이션 보안

코드 검토	**정기적인 코드 검토 및 보안 평가를 통해 취약점을 식별하고 완화**
종속성 관리	알려진 취약점을 보호하기 위해 도구를 사용하여 종속성을 스캔하고 업데이트
사고 대응 계획	보안 침해에 신속히 대응하고 피해를 완화하기 위해 사고 대응 계획을 개발하고 유지



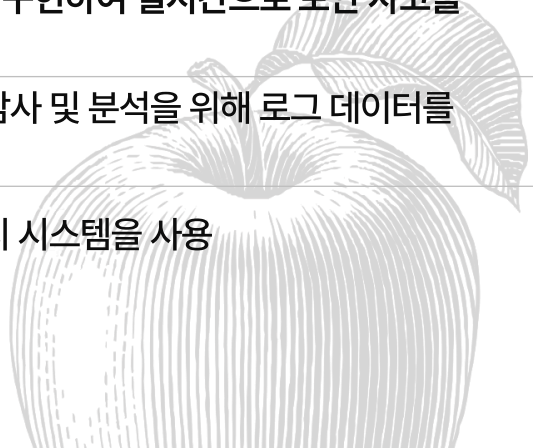
LLM 과 보안

7. 사용자 인증 및 권한 부여

강력한 인증	사용자 인증 보안을 강화하기 위해 다중 요소 인증(MFA)을 구현
세션 관리	세션 하이재킹 및 고정을 방지하기 위해 안전한 세션 관리 방법을 사용
권한 부여 검사	적절한 접근 제어를 보장하기 위해 권한 부여 검사를 정기적으로 검토하고 업데이트

8. 모니터링 및 로깅

지속적인 모니터링	애플리케이션 및 인프라의 지속적인 모니터링을 구현하여 실시간으로 보안 사고를 감지하고 대응
로그 관리	보안 관련 이벤트의 포괄적인 로깅을 보장하고 감사 및 분석을 위해 로그 데이터를 안전하게 저장
이상 탐지	비정상 패턴을 식별하고 대응하기 위해 이상 탐지 시스템을 사용



Part 6-6.
Evaluation

LLM 점수 매기고 줄 세우기

어려운 점 – 사람을 어떻게 평가하나?

- 시험점수로?
- 시험도 여러가지가 있다 - LSAT, GRE, 수능점수, 토플...
- 언어 모델 평가도 정말 여러가지
- 유튜브 구독자수?? 인스타 좋아요 수??

결론

- 한 가지 평가방법은 없음!
- 하지만 평가를 안 할 수는 없음!! 완벽하지는 않더라도 평가는 해야 퀄리티 컨트롤이 가능!
- 무엇에 이 모델을 쓸 것인지 해당 시나리오를 생각해야 함
  : 이것은 RAI, 보안, UI 등에도 도움이 됨!

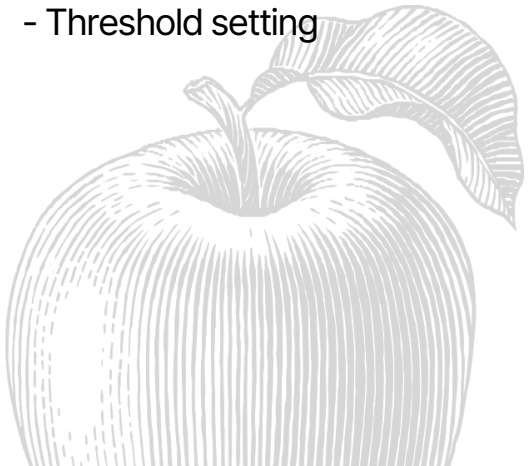
예시

목소리를 변환하여 텍스트로 저장

- 제대로 변환되었는가? → 이것은 테스트 가능!
- 요약을 해서 저장한다면, 요약이 제대로 되었는가? → 이것은 테스트 가능!

사용자 질문에 답할 수 있는 내용을 검색

- 질문을 검색에 용이하게 변환한다면, rewrite 의 품질 확인
  - Similarity?
  - Keyword search 로 검색 결과가 얼마나 나아졌는가
- 해당하는 답변을 성공적으로 찾아왔는가?
  - Precision, F1, recall...
- 해당하는 답변을 효율적으로 찾아왔는가?
  - Threshold setting



가져온 내용을 요약

- 요약이 제대로 되었는가?

예시

메타 프롬프트를 설정

- 프롬프트가 적절하고 효율적인가?
- 토큰수를 줄일 수 있는가?

최종 답변의 품질 확인

- 실제로 정확한 답변을 제공했는가?
- 가져온 데이터에서 답변을 제공했는가?

RAGAS 다시 재방문

Ragas score

generation

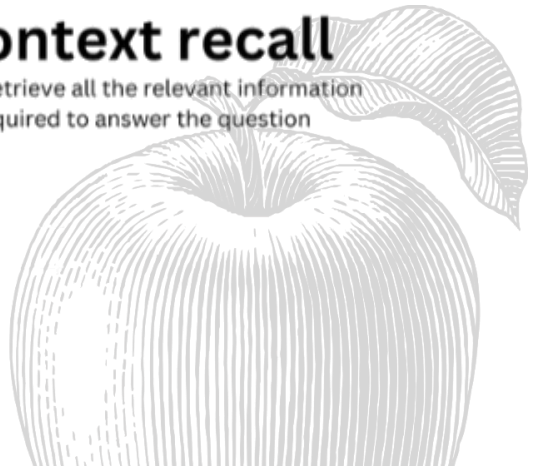
faithfulness
how factually acurate is
the generated answer

answer relevancy
how relevant is the generated
answer to the question

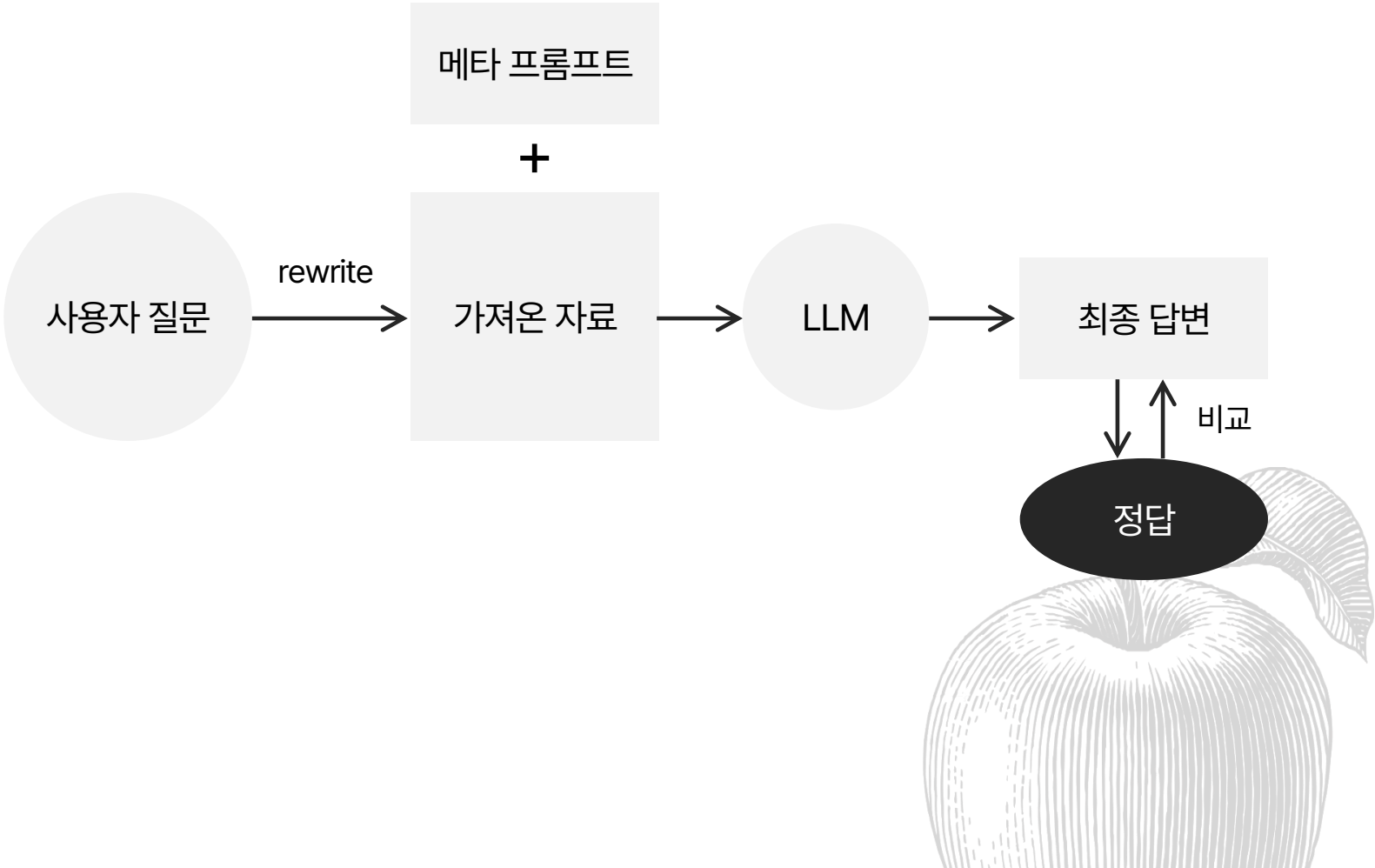
retrieval

context precision
the signal to noise ratio of retrieved
context

context recall
can it retrieve all the relevant information
required to answer the question



RAGAS 다시 재방문



RAGAS 다시 재방문

사용자가 이런 질문을 했는데, 내가 찾아보니까 이런 자료가 나왔다.
답변 좀 잘 해봐.

+

"프랑스 서울은 어디야?"

"프랑스, 수도"

1. 프랑스는 유럽의 국가이며,
2. 프랑스에서는 프랑스어를 합니다.
3. 프랑스는 음식이 맛있습니다.
4. 이번에 올림픽을 개최하네요.
5. 파리가 수도랍니다.

GPT4

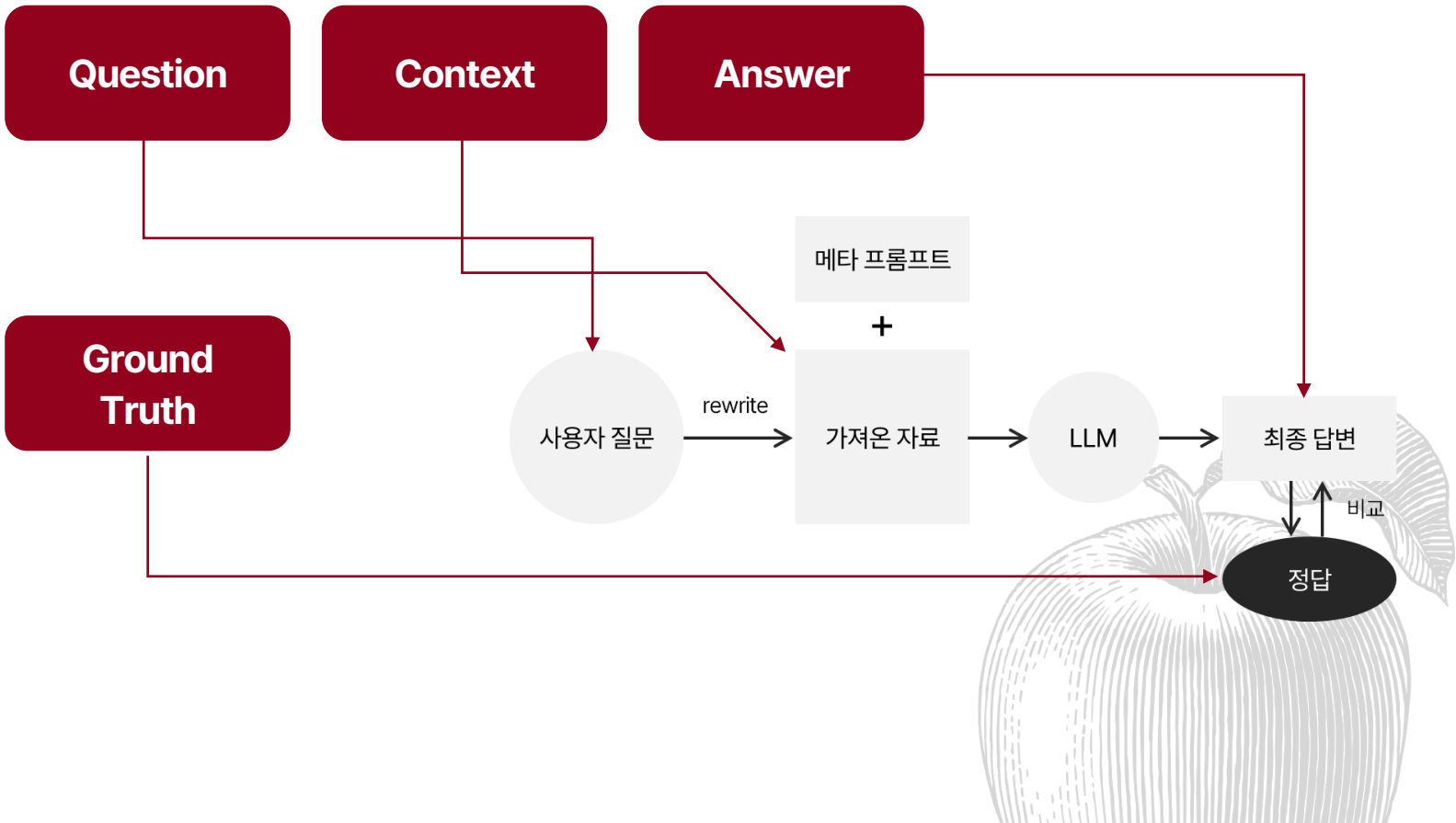
프랑스에는 서울이 없습니다. 서울은 한국의 수도지요. 수도를 말씀하신 거라면 프랑스의 수도는 파리입니다 이번해에 올림픽이 개최됩니다. 겁나 덥대요.

비교

프랑스의 수도는 파리 (Paris) 입니다.

RAGAS 평가에 필요한 요소

평가에 필요한 부분:



RAGAS metrics

- 얼마나 자료에 충실하게 정확하게 대답했는가
- 얼마나 질문에 맞는 대답을 했는가
- 얼마나 정답에 맞는 대답을 했는가
- 자료가 얼마나 도움이 되었는가
- 대답할 때 자료를 얼마나 사용했는가 (쓸데없이 많이 가져왔는가)
- 가져온 자료 중에 정답에 쓰인 부분이 얼마나 위에 위치했는가 (중요성 랭킹)
- ...

RAGAS metrics

질문	• 프랑스는 어디에 있는 나라고 수도는 어디인가?
컨텍스트 (가져온 자료)	• 유럽에는 여러 나라가 있고... • 프랑스는 요리로 유명하고... • 프랑스는 유럽에 있는 나라고... • 프랑스의 수도는 마르세이유이며...
답변	• 프랑스는 아프리카에 있는 나라이며 수도는 레이캬빅이다.
정답	• 프랑스는 유럽에 있는 나라이며 수도는 파리이다.

컨텍스트 거의 정확
답변은 틀림
정답과 다름

RAGAS metrics

질문	• 프랑스는 어디에 있는 나라고 수도는 어디인가?
컨텍스트 (가져온 자료)	• 유럽에는 여러 나라가 있고... • 프랑스는 요리로 유명하고... • 프랑스는 유럽에 있는 나라고... • 프랑스의 수도는 마르세이유이며...
답변	• 프랑스는 아프리카에 있는 나라이며 수도는 레이캬빅이다.
정답	• 프랑스는 아프리카에 있는 나라이며 수도는 레이캬빅이다.

컨텍스트는 거의 정확
답변은 틀림
정답과도 틀림
컨텍스트와도 틀림

실제 KPI

실제로 쓰는 metric:

Answer rate	얼마나 대답했는가 (“죄송합니다 모릅니다” 제외)
Fallback rate	얼마나 자료사용에 실패했는가
Satisfaction score	사용자의 피드백
Rephrase rate	똑같은 질문을 다시 물어본 %
Relevance	답변이 얼마나 연관성이 있는가
RAI / moderation rate	내용 센서 %
Security score	보안 스코어
Token count	얼마나 토큰을 많이 들였는가
Latency	얼마나 빨랐는가
Performance per data source	각 데이터 소스 마다 퍼포먼스

잘 알려진 metric

Precision, recall

Precision 정밀율	**Recall 재현율**
"죄인 중에 실제 죄인은 얼마?"	**"범죄자 중에 총 몇 명 잡았음?"**
죄 없는 사람 잡는 **노정밀 검사**	범죄자를 다 놓쳐서 범죄도시 재현하는 **노재현 검사**
백퍼센트 구형 선고 성공의 **최정밀 검사**	범죄자를 다 잡아내는 **최재현 검사**

잘 알려진 metric

Precision, recall

최정밀 + 노재현	실제 기소한 케이스는 정말 죄지은 사람이 많지만, 아예 잡지도 못한 범죄자들이 많음. 소수의 확실한 놈만 붙잡음. "왜 나만 잡아??"
노정밀 + 최재현	아주 많이 잡아들이는데 죄 없는 사람들도 많아서 실제 기소까지 못감 : "다 때려 잡아 넣는다고 해결되는 문제가 아니야!!"
최정밀 + 최재현	범죄가 일어나는 족족 100% 검거되며 억울한 사람은 1도 없음. 맥시멈 F1 스코어 1 의 위엄.

F1

- F1 점수=2×(정밀도+재현율/정밀도×재현율)
- F1 점수는 정밀도와 재현율 간의 균형을 맞춰야 할 때 특히 유용합니다.

잘 알려진 metric

BLEU (Bilingual Evaluation Understudy)

- 기계 번역의 품질을 평가하는 데 널리 사용되는 자동 지표

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

- 기계 번역, 요약, 및 자연어 생성(NLG) 시스템의 성능을 평가하는 데 널리 사용되는 자동화된 지표

수능 시험, 수학 시험, GRE, LSAT, SAT...

실제로 제일 많이 쓰이기: cosine similarity, custom score

Part 6-7.
RAI 해결하기

RAI와 규제

(변호사 필요하고 돈 들고 IT 도 아니고 까다롭고 돈 되는 것도 아닌데)
왜 해야 하는가???

영업 하다가 셧다운 할 수가 있음 (GDPR)

아예 영업이 불가능할 수가 있음 (HIPAA)

사실 이득이 되는 케이스 – Matzah*

- *마차는 유대교의 중요한 명절인 유월절(Passover) 동안 유대인들이 먹는 무교병입니다. 유월절은 고대 이스라엘 사람들이 이집트에서 탈출한 것을 기념하는 명절로, 이들은 급히 떠나야 했기 때문에 빵을 발효시킬 시간이 없었습니다. 그래서 무교병인 마차를 먹으며 그 역사를 기억합니다.
- *마차는 밀가루와 물로만 만들어지며, 발효 과정을 거치지 않기 때문에 평평하고 바삭한 식감을 가집니다. 전통적으로, 반죽에서 물과 밀가루가 혼합된 시점부터 굽는 과정까지 18분을 넘지 않아야 합니다. 이는 반죽이 자연적으로 발효되는 것을 방지하기 위함입니다.
- https://www.npr.org/sections/money/2012/04/10/150300040/why-matzo-makers-love-regulation

RAI 예시

RAI 임팩트 등급 평가

- 이 시스템은 무엇을 하는가
- 어떤 데이터를 가지고 무엇을 하고 누구에게 보여지는가
- 데이터 해킹의 가능성은?
- 딱 필요한 만큼만의 데이터를 사용하는가?
- 개인적인 정보를 이용하는가?
- 타인이 개인적인 정보를 볼 수 있는가?
- 타인이 이 시스템의 결과물로 해를 입힐 수 있는가?
- 문제가 생길 경우에 어떤 방안이 준비되어 있는가
- 문제가 생길 경우 최악의 임팩트는 무엇인가?
- 공정하고 윤리적인가?
- 책임자는 누구인가
- Red team 테스트는 어떻게 시행되었고 결과는 무엇이었는가?
- 어느 지역에서 어떤 언어로 서비스가 되는가
- RAI / 보안 위협을 지속적으로 체크하고 있는가?
- 체크하는 방법과 threshold 가 무엇인가?

보안팀 체크

- 법무부 팀 및 해당 지역 책임자 체크

RAI 예시

꼭 다 해야 하는가?

- 법적으로 문제 될 수 있는 부분이라면
- 프로덕트의 reliability 와 생존이 걸린 문제
- 회사의 평판이 걸린 문제
- 그 후로는 최대한 효율적으로, 배보다 배꼽이 더 크지 않게

RAI 툴:

- Azure RAI dashboard
- Google RAI
- Moderation - OpenAI

하나하나의 툴도 중요하지만 전체적인 시각이 필요함

RAI 는 회사의 생존과 프로덕트의 성공을 위해서 꼭 필요함

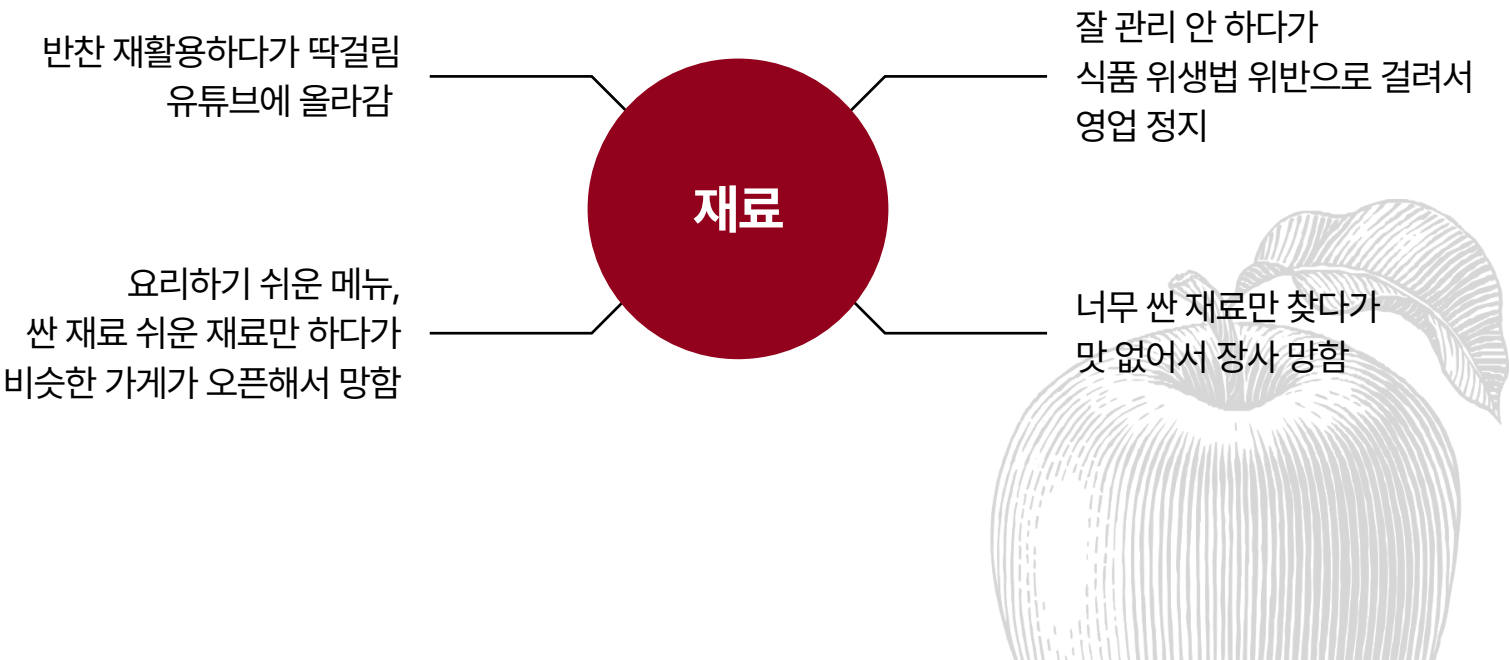
Part 6-8.
데이터 프라이버시, 데이터 관리, 윤리적 문제, 퀄리티 컨트롤

Data privacy, governance

어디선가 계속 들었던 것 같은 이 느낌 뭐지...?

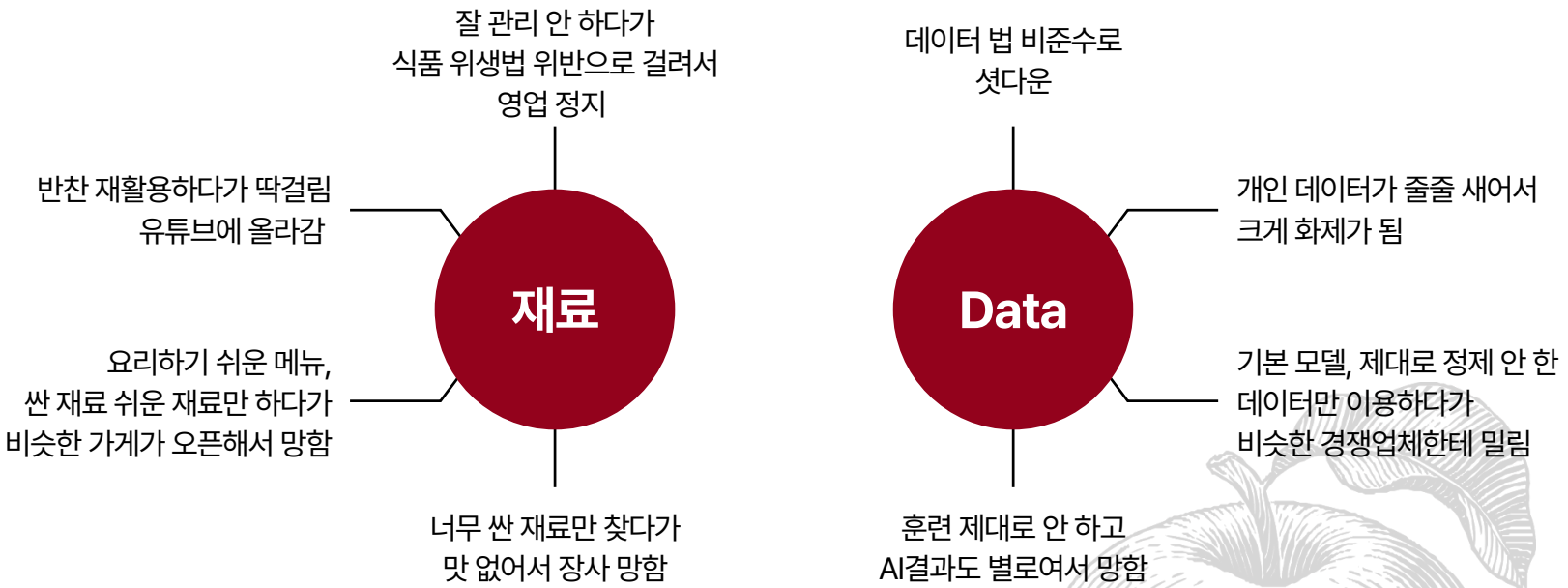
우리 바로 앞에서도 RAI 얘기하면서 데이터 얘기하지 않았나요??

데이터 얘기 왜 자꾸 나오나요 보안이랑 RAI 랑 섞여서 나오나요...?



Data privacy, governance

보안, 윤리, 관리 등이 다 엮여 있음.



하지만 지금 꼭 해야 하나요...?

프로젝트 규모에 따라 다름 (포장마차 vs 종가집 김치공장)

- 작은 스타트업?
- 회사 주요 프로덕트?
- 국제적 서비스?

회사 규모에 따라 다름

- 정부 고객 상대
- 공개적으로 잘 알려져 있는 고객 상대

통합해야 하는 시스템에 따라 다름

**하지 않더라도 최소한 주로 어떻게 관리한다,
어떤 것부터 신경 써야 한다는 알아두는 것이 좋음**

가장 중요

LLM 쿼리 데이터

- "제가 이런 죄를 지었는데 처벌 피하는 것 가능할까요...?" - 모 유명 연예인

RAG 시스템에서 데이터를 어떻게 가지고 오는가

- Auth, 데이터 유출 방지

  의료 시스템
  - 의료 스태프와 환자들이 접근할 수 있는 데이터 제한.
  - 참고 자료 (context)에서 개인 정보 유출이 되지 않도록 함 (fine tuning 된 자료)
  - "다른 환자분 이모씨(53세) 는 6개월 전에 치매 진단을 받고...

GDPR 준수

프로덕션 데이터 시스템 데이터 보호

- Facebook 에서 전애인의 데이터를 뒤지는 경우?

가장 중요

윤리 - 해킹당한 챗봇이

- 인종차별 발언, 음담패설, 정치적인 발언

윤리 -

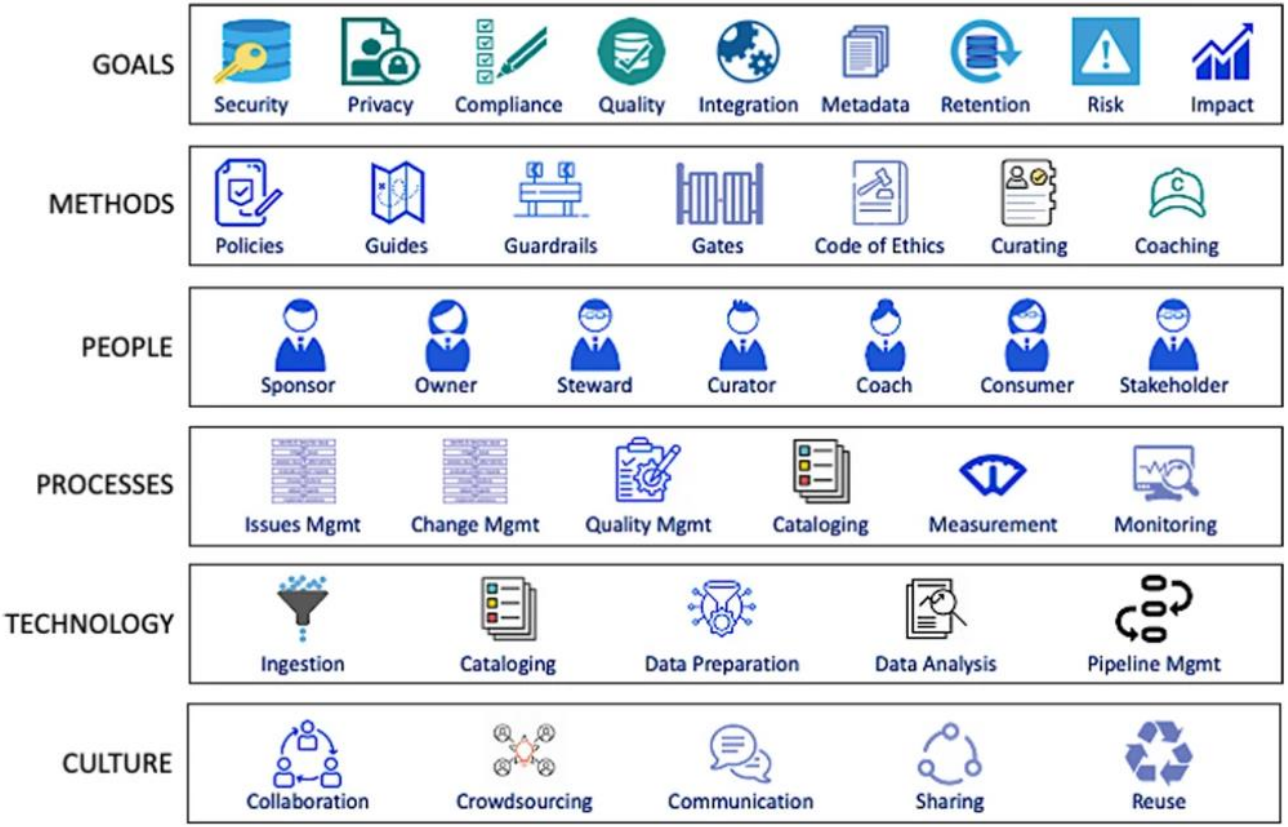
- 흑인에게는 융자 안 주는 시스템
- 동양인 얼굴을 늘 비슷한 얼굴로 만드는 AI
- 백인은 모두 북미 국적으로 디폴트 값을 세팅하는 AI
- 인도계의 가상 인물을 늘 특정 카스트의 성과 피부톤으로 세팅하는 AI

소비자의 개인 정보 유출

실제 케이스 -

- 구글: “미국 여자 이미지를 만들어줘” - 흑인으로만 만듬
- Tay: 트위터 데뷔 16시간 내에 퇴장

기본 frameworks



데이터 stage 마다 체크

훈련용 데이터 체크	• 윤리적인가 • 정확한가 • 편견이 없는가 • 개인 정보가 없는가
추론 결과 체크	
사용자 질문 체크	
답변 체크	
로깅 데이터 체크	

어떻게 통제할 것인가

- 잴 수 없으면 모니터할 수 없다
- 프로세스가 없으면 저절로 관리되지 않는다
- 하지만 배보다 배꼽이 클 수는 없다
- Cover the basics, then pick your battles

Part 6-9.
여러가지 툴 돌아보기

툴 타입

RAI

Data

Security

Moderation

Evaluation

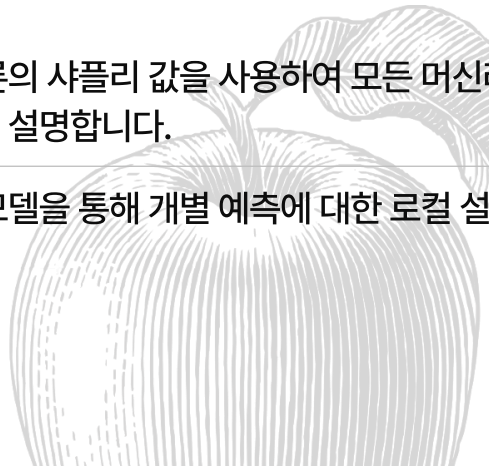
RAI

1. 공정성, 편향 탐지/완화 도구

Fairlearn	Microsoft에서 제공하는 오픈 소스 툴킷으로, 머신러닝 모델의 공정성을 평가하고 개선하기 위한 알고리즘과 지표를 제공합니다.
Aequitas	다양한 그룹 간의 편향을 평가하는 편향 및 공정성 감사 도구입니다.
IBM AI Fairness 360 (AIF360)	머신러닝 모델의 편향을 점검하고 이를 완화하기 위한 지표와 알고리즘을 제공하는 오픈 소스 라이브러리
Google's What-If Tool	모델 성능을 조사하기 위한 인터랙티브 비주얼 도구로, 편향 분석 기능을 포함합니다.

2. 설명 가능성 도구

SHAP (SHapley Additive exPlanations)	협력 게임 이론의 샤플리 값을 사용하여 모든 머신러닝 모델의 출력을 설명합니다.
LIME (Local Interpretable Model-agnostic Explanations)	해석 가능한 모델을 통해 개별 예측에 대한 로컬 설명을 제공합니다.



RAI

7. 윤리적 AI 프레임워크 및 가이드라인

European Commission's Ethics Guidelines for Trustworthy AI	신뢰할 수 있는 AI를 위한 윤리적 지침.
IEEE's Ethically Aligned Design	윤리적으로 정렬된 설계를 위한 IEEE 표준.
Google's AI Principles	Google의 AI 원칙.

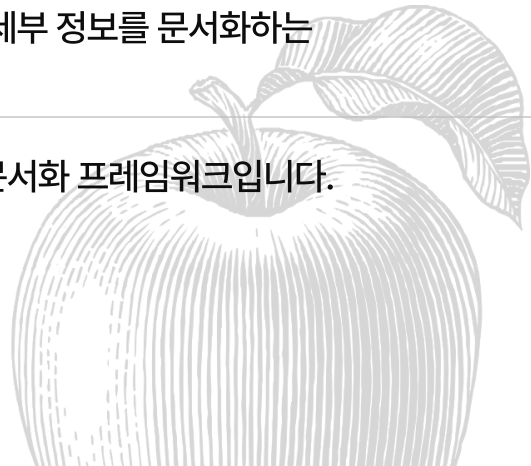
데이터

3. 프라이버시 도구

Differential Privacy Libraries	개별 데이터 포인트를 보호하기 위해 차등 프라이버시 기술을 구현하는 라이브러리들입니다.
OpenMined PySyft	안전하고 프라이빗한 딥 러닝을 위한 프레임워크입니다. (링크: https://github.com/OpenMined/PySyft)
TensorFlow Privacy	차등 개인 정보 보호 기능을 사용하여 머신러닝 모델을 훈련하기 위한 라이브러리입니다. (링크: https://github.com/tensorflow/privacy)

5. 책임 및 투명성 도구

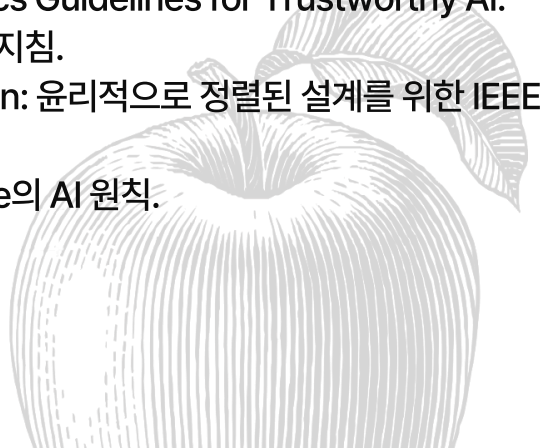
Model Cards	델의 사용 용도, 성능, 제한 사항 등에 대한 세부 정보를 문서화하는 프레임워크입니다.
Datasheets for Datasets	데이터셋에 대한 투명성을 제공하기 위한 문서화 프레임워크입니다.



전체적인 거버넌스

8. 거버넌스 및 컴플라이언스 도구

Azure Machine Learning's Responsible AI dashboard	공정성 평가, 오류 분석, 해석 가능성을 위한 도구를 제공합니다.
AI Fairness Checklist	AI 시스템의 공정성을 평가하기 위한 체크리스트입니다.
AI Incident Database	AI 시스템과 관련된 사고 데이터베이스입니다. (링크: https://incidentdatabase.ai/)
Partnership on AI	책임감 있는 AI 개발을 위한 모범 사례를 개발하고 공유하는 다중 이해관계자 조직입니다. (링크: https://www.partnershiponai.org/) • European Commission's Ethics Guidelines for Trustworthy AI: 신뢰할 수 있는 AI를 위한 윤리적 지침. • IEEE's Ethically Aligned Design: 윤리적으로 정렬된 설계를 위한 IEEE 표준. • Google's AI Principles: Google의 AI 원칙.



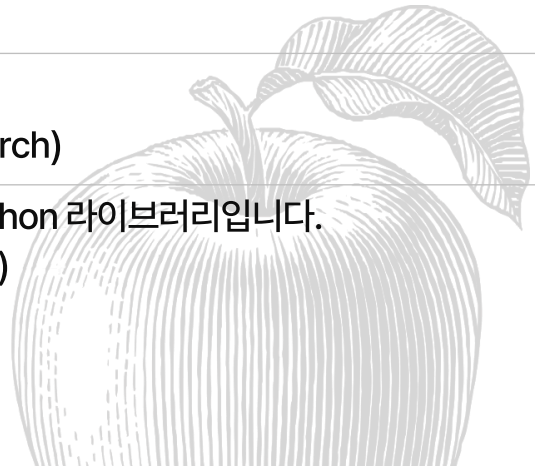
보안

1. 보안 도구

Adversarial Robustness Toolbox (ART)	IBM에서 제공하는 오픈 소스 라이브러리로, 적대적 공격에 대한 모델 방어를 돕습니다.
CleverHans	적대적 예제에 대한 머신러닝 모델의 취약성을 평가하는 라이브러리입니다.

5. 책임 및 투명성 도구

Model Cards	모델의 사용 용도, 성능, 제한 사항 등에 대한 세부 정보를 문서화하는 프레임워크입니다.
TextAttack	NLP에서 적대적 공격, 데이터 증강 및 모델 훈련을 위한 Python 프레임워크입니다. (링크: https://github.com/QData/TextAttack)
Advertorch	적대적 견고성 연구를 위한 도구 상자입니다. (링크: https://github.com/BorealisAI/advertorch)
Foolbox	머신러닝 모델을 속이는 적대적 예제를 만드는 Python 라이브러리입니다. (링크: https://github.com/bethgelab/foolbox)



보안

모니터링 및 감사 도구

Arthur AI	프로덕션 환경에서 머신러닝 모델의 성능을 모니터링하고 개선하기 위한 플랫폼입니다.
Adversarial Robustness Toolbox (ART)	IBM에서 제공하는 오픈 소스 라이브러리로, 적대적 공격에 대한 모델 방어를 돕습니다.
CleverHans	적대적 예제에 대한 머신러닝 모델의 취약성을 평가하는 라이브러리입니다
WhyLabs	머신러닝 애플리케이션의 데이터 및 모델 상태를 모니터링하기 위한 플랫폼입니다.

Part 6-10.
프론트엔드 최적화

LLM 과 프론트엔드

왜??? 프론트엔드에 걱정해야 하지???

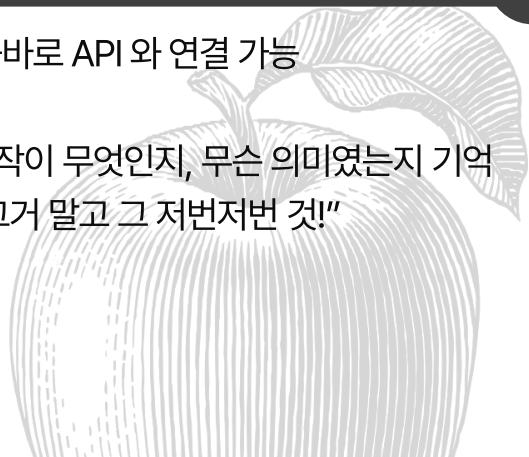
현재의 AI: 사람의 "의도"를 이해하고, 알아서 "백엔드"를 실행할 수 있음

이전의 시스템:

- 모든 내용을 포함한 "메뉴"를 만들고, "정확한 단어 검색"을 제공
- step-by-step workflow 를 제공

새로운 AI 인터페이스

- 사람의 말을 이해하고 곧바로 API 와 연결 가능
- 멀티 모달
- iteration - 아까 했던 동작이 무엇인지, 무슨 의미였는지 기억
  "아니 가까 그거, 아니 그거 말고 그 저번저번 것!"



LLM 과 프론트엔드

"의도"를 이해하고, 알아서 "백엔드"를 실행할 수 있음

최적화 목표

- 빨리 의도를 파악

조심해야 할 부분

- 워크플로우를 보기 힘듬
- 세션 흐름이 없으므로 현존하는 플로우에 플러그인 하기가 힘들 수 있음
- 음성/이미지등 멀티 모달 핸들링

LLM 과 프론트엔드

- Streaming output
- 시각화 컨트롤 (이미지 아웃풋)
- 이미지 가격 / 스피드
- 의인화 문제
- 세션을 어디까지 저장할 것인가
- 챗창이 어디에나 있을 경우 컨텍스트 이해 및 데이터를 얼마나 전송할 것인가
- 빠른 응답 시간
- 싱글 턴, 멀티 턴
- 프롬프트를 제공
- deflection, alternative offering
- 보안 관리

LLM 과 프론트엔드

https://www.microsoft.com/en-us/haxtoolkit/library/?content_type%5B0%5D=guideline&content_type%5B1%5D=pattern

AI 기능 발견할 수 있게 하기	**구글 AI Overview, Copilot button, bing**
무엇을 할 수 있는지 확실하게 말하기	
샘플을 보여주기	옵션 두세개 주기 (구글 맵 선택, 디자인 아이디어)
제한을 확실히 하기	"불확실할 수 있습니다", 구글 서치 "It may be incorrect
컨텍스트에 맞게 하기	스크린에 맞는 AI, 흐름에 맞는 AI
쉽게 찾고 쉽게 없앨 수 있게 하기	
쉽게 고칠 수 있도록 하기	
왜 고쳤는지 설명할 수 있도록 하기	
피드백 받기	
최근 대화 내용 기억하기	
계속 업데이트하기	

Part 6-12.

LLM 시스템 최적화란?

LLM 시스템의 최적화

언제나 최우선은 생존!

프로세스가 먼저냐 프로덕트가 먼저지...

- 시스템 개발의 최적화
- LLM Ops
- 스피드/가격 통제
- CI/CD
- 보안
- Evaluation
- RAI
- Data
- Front-end

LLM 시스템의 최적화

언제나 최우선은 생존!
프로세스가 먼저냐 프로덕트가 먼저지...

제품 개발을 빠르게, 안전하게, 위험 낮게,
저렴하게, 효율적으로, 관리하기 쉽게!

Part 6-12.

Azure/AWS 클라우드
- Tech stack

Tech Stack sample

Tech stack sample

- AWS
  - https://aws.amazon.com/blogs/containers/build-generative-ai-apps-on-amazon-ecs-for-sagemaker-jumpstart/
- Azure
- Google
- Lightning.AI

그 외 유명한 툴+서비스