▣ **강의명:** CSCI E-103: 재현 가능한 머신러닝

▣ **주차:** Lecture 11

▣ **교수명:** Anindita Mahapatra

Eric Gieseke

▣ **목적:** Lecture 11의 핵심 개념 학습

Contents

1 개요: AI의 진화와 LLM 개발의 현재

이 문서는 대규모 언어 모델(LLM)과 에이전트(Agent)를 활용하여 실제 비즈니스 가치를 창출하는 방법을 다룹니다. 단순히 "신기한 기술"을 넘어, 데이터를 통합하고 배포하며 운영하는 전체 수명주기(LLMOps)를 이해하는 것이 목표입니다.

□ 핵심 정보

학습 목표

- **이론:** AI/ML의 진화 과정과 생성형 AI(GenAI)의 차별점 이해
- **전략:** 조직의 성숙도에 따른 LLM 도입 단계 (Prompting → RAG → Fine-tuning)
- **실무:** Databricks의 도구(Genie, Agent Bricks, Unity Catalog)를 활용한 에이전트 구축
- **운영:** LLMOps, 환각(Hallucination) 관리, 윤리적 AI 사용

2 필수 용어 정리

생성형 AI 분야는 새로운 용어가 많습니다. 아래 표는 이 문서를 이해하기 위한 핵심 단어장입니다.

용어 (약어)	원어	쉬운 설명 (직관적 비유)
LLM	Large Language Model	인터넷의 모든 텍스트를 읽고 언어 패턴을 익힌 **'초거대 독서광 AI'**
GenAI	Generative AI	데이터를 분류하는 것을 넘어, 새로운 텍스트/이미지를 **'창조'**하는 AI
Hallucination	Hallucination	AI가 사실이 아닌 내용을 마치 사실인 것처럼 **'뻔뻔하게 거짓말'**하는 현상
RAG	Retrieval Augmented Generation	AI에게 **'오픈북 테스트'**를 치르게 하듯, 참고 자료를 검색해서 보여주고 답하게 하는 기술
Grounding	Grounding	AI의 답변을 현실 세계의 사실이나 데이터에 **'단단히 묶어두는'** 과정
Fine-Tuning	Fine-Tuning	일반적인 지식을 가진 AI에게 특정 분야(예: 법률, 의학)를 **'심화 과외'** 시키는 것
Embeddings	Embeddings	텍스트의 의미를 컴퓨터가 이해할 수 있는 **'숫자 좌표(벡터)'**로 변환하는 기술
Vector DB	Vector Database	의미(Embedding)를 저장하여, 키워드가 달라도 **'문맥상 유사한'** 자료를 찾아주는 검색 엔진
Chain of Thought	Chain of Thought	AI에게 "답만 말하지 말고 **'풀이 과정'**을 단계별로 써라"고 시키는 기법

Table 1: *LLM* 및 생성형 *AI* 핵심 용어 요약

3 핵심 개념: AI의 진화와 도입 전략

3.1 1. AI의 포함 관계 (Venn Diagram)

AI 기술은 갑자기 튀어나온 것이 아니라, 수십 년간 발전해 온 기술의 집약체입니다.

- **AI (Artificial Intelligence):** 인간의 지능을 모방하는 모든 시스템 (가장 큰 범주)
- **ML (Machine Learning):** 데이터를 통해 학습하는 AI의 하위 분야
- **Deep Learning (Neural Networks):** 인간의 뇌 신경망을 모방한 복잡한 ML
- **GenAI (Generative AI):** 기존 데이터를 분석하는 것을 넘어, 새로운 데이터를 생성하는 딥러닝의 최신 분야
  - **LLM:** 텍스트 생성 특화 (예: GPT, Gemini, Claude)
  - **GAN:** 이미지/비디오 생성 특화 (예: Deepfake, StyleGAN)

3.2 2. LLM 성숙도 모델 (Maturity Curve)

기업이나 개인이 LLM을 도입할 때, 무조건 처음부터 자체 모델을 만드는 것은 비효율적입니다. 비용과 복잡도를 고려하여 단계별로 접근해야 합니다.

LLM 도입의 4단계 (쉬움 → 어려움)

1. **Prompt Engineering (프롬프트 엔지니어링)**
   - **설명:** AI에게 일을 시키는 명령어를 정교하게 다듬는 단계.
   - **비용/난이도:** 매우 낮음 / 데이터 필요 없음.
   - **한계:** AI가 학습하지 않은 최신 정보나 사내 비공개 정보는 모름.
2. **RAG (검색 증강 생성) - *가장 권장되는 단계**
   - **설명:** 사내 문서나 데이터베이스를 검색(Retrieval)하여 그 내용을 AI에게 참고자료로 주고 답변(Generation)하게 함.
   - **장점:** 최신 정보 반영 가능, 할루시네이션 감소, 가성비 최고.
   - **비유:** 시험 보는 학생(AI)에게 교과서(사내 데이터)를 펼쳐놓고 답을 쓰게 하는 것.
3. **Fine-Tuning (파인 튜닝)**
   - **설명:** 기존 모델에 우리만의 데이터로 추가 학습을 시키는 것.
   - **용도:** 특수한 말투, 전문 용어, 특정 형식의 답변이 필요할 때.
   - **비용:** 데이터 준비와 학습에 상당한 비용 발생.
4. **Pre-Training (사전 학습)**
   - **설명:** 처음부터 모델을 바닥부터 만드는 것 (예: 자체 GPT 만들기).
   - **현실:** 구글, 메타급 기업이 아니면 비용(수십 수백 억 원) 문제로 거의 불가능.

4 기술 심화: RAG (검색 증강 생성)

LLM의 가장 큰 약점인 **"최신 정보 부재"**와 **"할루시네이션(거짓말)"**을 해결하는 가장 현실적인 기술입니다.

4.1 1. 왜 RAG가 필요한가?

LLM은 학습이 끝난 시점(예: 2023년) 이후의 정보는 모릅니다. 또한, 우리 회사의 비공개 문서는 당연히 본 적이 없습니다. 이를 해결하기 위해 모델을 재학습시키는 것은 너무 비쌉니다. RAG는 **"외부 지식 검색"**을 통해 이 문제를 해결합니다.

4.2 2. RAG의 작동 원리 (Workflow)

RAG 시스템은 크게 **데이터 준비(Ingestion)**와 **검색 및 답변(Retrieval & Generation)** 두 단계로 나뉩니다.

- **단계 1: 데이터 준비 (사전 작업)**
  1. 문서를 잘게 쪼갭니다 (Chunking). (예: 800 1200자 단위)
  2. 쪼갠 문서를 AI가 이해하는 숫자 배열(Vector/Embedding)로 변환합니다.
  3. 이 숫자를 **Vector DB**에 저장합니다. (유사도 검색을 위해)
- **단계 2: 실제 질문 시 (실시간)**
  1. 사용자가 질문을 합니다. ("우리 회사의 재택근무 규정이 뭐야?")
  2. 질문을 숫자(Vector)로 변환합니다.
  3. Vector DB에서 질문과 숫자가 가장 비슷한(의미가 유사한) 문서 조각을 찾아냅니다.
  4. 찾아낸 문서 조각을 **프롬프트에 붙여서** LLM에게 보냅니다.
  5. LLM은 "아래 문서를 참고해서 답변해"라는 지시를 받고 정확한 답을 생성합니다.

초심자가 자주 하는 오해 **Q. RAG를 쓰면 LLM이 학습을 하나요?**
A. 아닙니다! RAG는 LLM에게 잠시 참고자료를 보여주는 것뿐입니다. LLM의 뇌(모델 파라미터) 자체는 변하지 않습니다. 마치 시험 때 오픈북을 한다고 해서 학생의 지능이 영구적으로 변하는 것은 아닌 것과 같습니다.

5 실무: Databricks 도구를 활용한 구현

Databricks 플랫폼은 LLM 애플리케이션 개발을 위한 통합 도구를 제공합니다.

5.1 1. Genie (지니)

- **정의:** 정형 데이터(SQL 테이블)와 대화하는 에이전트.
- **특징:** 텍스트로 질문하면 자동으로 SQL 쿼리를 생성하여 답을 줍니다.
- **장점:** 데이터의 메타데이터(스키마)만 보고 SQL을 짜기 때문에, 일반 LLM보다 할루시네이션이 적고 정확도가 높습니다.
- **사용 예:** "지난달 매출이 가장 높은 제품 5개 보여줘" → `SELECT name, sales FROM ...` 자동 실행

5.2 2. Agent Bricks (에이전트 브릭스)

- **정의:** 코딩 없이 클릭 몇 번으로 RAG 에이전트를 만드는 'AutoML' 같은 도구.
- **기능:** PDF 파일이 있는 경로만 지정하면, 자동으로 벡터 DB를 구축하고 챗봇을 생성해줍니다.
- **활용:** 고객 응대 챗봇, 사내 규정 검색기 등을 10분 만에 프로토타이핑 가능.

5.3 3. AI Functions (SQL에서 AI 쓰기)

복잡한 파이썬 코드 없이, SQL 쿼리 안에서 바로 LLM 기능을 호출할 수 있습니다.

```
-- 고객리뷰의감정을분석하고요약하는       SQL 쿼리
SELECT
    review_text,
    ai_analyze_sentiment(review_text) AS sentiment, -- 긍정부정/ 분석
    ai_summarize(review_text, 10) AS summary        -- 단어10 요약
FROM customer_reviews
WHERE date > '2025-01-01';
```

Listing 1: AI Functions 활용 예시

5.4 4. LLMOps (LLM 운영)

기존 MLOps(머신러닝 운영)에 **"언어적 특성"**이 추가된 개념입니다.

- **모델 평가:** 답변이 정확한지, 독성이 없는지 평가 (LLM을 심판으로 사용하기도 함).
- **피드백 루프 (RLHF):** 사용자의 "좋아요/싫어요" 버튼 클릭을 모아 모델을 개선.
- **보안:** 프롬프트 인젝션(해킹) 방지, 개인정보 유출 방지.

6 운영과 윤리: 리스크와 대응

LLM은 강력하지만 위험 요소도 큽니다. 이를 관리하는 것이 엔지니어의 핵심 역량입니다.

리스크 유형	설명	대응 방안
환각 (Hallucination)	없는 사실을 지어냄	RAG 사용, Temperature(창의성 수치)를 0으로 설정
편향 (Bias)	특정 인종/성별에 차별적 발언	학습 데이터의 다양성 확보, 필터링 가드레일 설치
보안 위협	프롬프트 인젝션 (AI 속이기)	입력값 검증, 관리자 권한 분리
개인정보 유출	학습된 민감 정보 노출	학습 데이터 정제(Masking), 사내 전용 모델 사용

Table 2: *LLM* 주요 리스크 및 대응 방안

□ 핵심 정보

ESG와 AI 최근 기업은 **ESG(환경, 사회, 지배구조)** 관점에서 AI를 평가합니다. AI 모델 학습에 막대한 전기가 소모(환경)되거나, AI가 차별적 발언(사회)을 하는 것은 기업 가치에 치명적입니다. 따라서 "윤리적 AI"는 선택이 아닌 필수입니다.

7 학습 체크리스트 및 공지사항

7.1 주요 일정 (강의 공지)

- **성적 공개:** Quiz-1, Assignment-3 채점 완료.
- **과제:** Use Case-1, Assignment-4 곧 채점 예정.
- **최종 프로젝트:**

  - 11월 18일: 프로젝트 주제 및 세부 내용 공개.
  - 12월 16일: 최종 발표 (팀원 전원 참석 권장).
- **기타:** 다음 주는 추수감사절(Thanksgiving)로 휴강. 12월 9일 산업계 전문가 초청 강연.

7.2 FAQ: 자주 묻는 질문

Q. RAG를 쓸 때, 문서가 업데이트되면 어떻게 하나요?
A. Vector DB를 업데이트해야 합니다. 문서를 통째로 교체하거나 변경된 부분만 다시 임베딩하여 덮어써야 합니다. 다만, 부분 업데이트 시 버전 관리가 복잡해질 수 있습니다.

Q. LLM이 항상 최신 정보를 알 수 있나요?
A. 순수 LLM은 불가능합니다. RAG나 웹 검색 에이전트를 붙여야만 실시간 정보를 반영할 수 있습니다.

Q. 프롬프트 엔지니어링만으로 충분한가요?
A. 간단한 업무는 충분합니다. 하지만 전문적인 지식이 필요하거나 사내 데이터를 써야 한다면 RAG가 필수적입니다.

8 빠르게 훑어보기: 1페이지 핵심 요약

□ 핵심 요약

1. **AI의 흐름:** Rule-based → ML → Deep Learning → **GenAI(생성형 AI)**.
2. **LLM 활용 전략:** 무작정 모델을 만드는(Pre-train) 것이 아니라, **RAG(검색 증강)**를 통해 사내 데이터를 연결하는 것이 가장 효율적임.
3. **RAG 핵심:** 문서 쪼개기(Chunking) → 숫자 변환(Embedding) → Vector DB 저장 → 유사도 검색 → 답변 생성.
4. **Databricks 도구:**
   - **Genie:** 정형 데이터(SQL) 담당, 정확도 높음.
   - **Agent Bricks:** 비정형 데이터(PDF 등) 담당, 빠른 구축.
   - **AI Gateway:** 모델 보안 및 관리의 관문.
5. **주의사항:** 할루시네이션(거짓말) 방지, 데이터 보안, 윤리적 책임이 기술 구현만큼 중요함.
6. **미래:** 단순 챗봇을 넘어, 스스로 도구를 선택하고 행동하는 **에이전트(Agent)** 시대로 진입 중.

부록: 초심자를 위한 Agent 구축 단계 (Pseudo-Flow)

Databricks에서 RAG 에이전트를 만드는 사고의 흐름입니다.

1. **데이터 준비:** PDF 파일들을 Unity Catalog Volume에 업로드.
2. **인덱싱 (Vector Search):** "이 파일들을 읽어서 검색 가능하게 만들어줘" (Agent Bricks 활용 시 클릭 몇 번으로 완료).
3. **에이전트 생성:** "이 인덱스를 참고하는 챗봇을 만들어."
4. **테스트 (Playground):** 채팅창에서 질문해보고 답변 확인.
   - 답변이 이상하다? → 프롬프트 수정 ("전문가처럼 답변해", "모르면 모른다고 해").
   - 근거가 없다? → 데이터(PDF) 보강.
5. **배포:** Review App을 통해 사용자들에게 링크 공유 및 피드백 수집.