

? LLM ?? ?????

? LLM???

LLM(Large Language Model) 이 '언어 모델' 이라는 의미로, '대형 언어 모델' 이라고도 불리며, AI의 한 분야인 자연어 처리(NLP)에 속한다.

? ??? ??

LLM은 방대한 양의 텍스트 데이터를 학습하여, 입력된 텍스트에 대해 맥락적으로 적절한 다음 단어를 생성하는 모델이다. 이는 번역, 요약, 생성, 대화 등 다양한 NLP 작업을 수행하는 데 사용된다.

주요 LLM: ChatGPT, Claude, Gemini 등

?? LLM? ??? ?????

LLM은 크게 3가지 유형으로 나뉘며, 각각의 특징은 다음과 같다.

1?? ?? (Training)

1. 데이터 수집: 방대한 양의 텍스트 데이터를 수집한다.

- 데이터는 다양한 출처(책, 웹, 뉴스, 논문 등)에서 수집된다.
- 데이터는 토큰화(Tokenization) 과정을 거친다.
- "이 문장은 매우 흥미로운 주제에 대해 논쟁을 일으켰다"와 같은 문장을 예로 들 수 있다.

2. 학습:

```

입력 프롬프트: "이 문장은 매우 흥미로운 주제에 대해 논쟁을 일으켰다"
→ LLM 출력: "이 문장은 매우 흥미로운 주제에 대해 논쟁을 일으켰다"
→ "이 문장은 매우 흥미로운 주제에 대해 논쟁을 일으켰다"
  
```

2?? ?? (Understanding)

3. 평가: 모델의 성능을 평가한다.

이것은 매우 흥미로운 주제이며, 많은 사람들이 궁금해하는 주제입니다.

예 :

```
사용자: "이것이 무엇인가요?"
→ LLM 사용자:
- "이것" = 무엇 (이것이 무엇인가요? 이것이 무엇!)
- "이것이" = 이것이 무엇
- 이 것이: 이 것이 무엇 이 것이 무엇 이 것이 무엇
```

3?? ?? (Generation)

예 : 이 것이 무엇 이 것이 무엇 이 것이 무엇 이 것이 무엇

이것이 무엇 이것이 무엇 이것이 무엇 이것이 무엇 이것이 무엇 이것이 무엇 이것이 무엇

예 :

```
사용자: "이것이 무엇인가요?"

LLM 사용자:
1. "이것이" → 이 것이 무엇 → "이" (이 것이)
2. "이" → 이 것이 무엇 → "이것이" (이 것이)
3. "이것이" → 이 것이 무엇 → "이것이" (이 것이)
4. "이" → 이 것이 무엇 → "이것이" (이 것이)

LLM 사용자: "이것이 무엇 이것이 무엇"
```

? LLM ??? ??: ??? ??

LLM은 매우 흥미로운 주제이며, 많은 사람들이 궁금해하는 주제입니다.

예 : 이 것이 무엇 이 것이 무엇 이 것이 무엇
예 : 이 것이 무엇 이 것이 무엇 이 것이 무엇 (이 것이) △

? ??? ??

[]

↓

[] → " " + " " + " " + " " + "?"

↓

[LLM]

- ()
-
-

↓

[]

- " " (85%)
- " " (3%)
- " " (0.1%)

↓

[] → " "

↓

[] → " " → ",."

↓

[] " ."

? LLM? ??

? ??

-
-
-
-

?? ??

- (: 2024 2025)
-
- (, Hallucination)
-
-

? ????

? RAG: LLM?? ?? ??? ??

? LLM? ??

LLM은 텍스트, 이미지, 오디오 등을 생성할 수 있다.

? ?? ?? ??

입력 프롬프트	LLM 출력	참고 사항
"2024년 3월 31일 현재 상황은?"	현재는 2024년 3월 31일입니다.	현재 날짜를 출력함
"API 호출 방법 알려주세요"	API 호출 방법은 다음과 같습니다.	API 호출 방법을 설명함
"이 텍스트를 요약해주세요"	요약된 텍스트입니다.	텍스트를 요약함
"이 텍스트를 번역해주세요"	번역된 텍스트입니다.	텍스트를 번역함

? RAG?? (Retrieval-Augmented Generation)

LLM은 외부 데이터베이스에서 정보를 검색하여 생성할 수 있다.

? ? ??????

1. 입력: LLM은 외부 데이터베이스 (예: 문서, 웹 페이지)에서 정보를 검색한다.

2. 출력: 검색된 정보를 바탕으로 LLM은 질문에 대한 답변을 생성한다.

?? ?? ??

1. 입력: "2024년 3월 31일 현재 상황은?"

↓

2. RAG 출력: 검색된 정보를 바탕으로 "2024년 3월 31일 현재 상황은"에 대한 답변을 생성함

↓

3. RAG 000: 00 00 000 000 LLM00 00

↓

4. LLM: 00 000 00 000000 00 00

↓

5. 000: "300 000 12000000 00 00 15% 000000"

? ??? ??

000 00 000 0000 000 .

- 00 LLM: 000 00 0000 00 00
- RAG 00 LLM: 00 0 0 0000 000 0000 0 00

?? ??? ??

? ??

- 00 00 00 : 000 000000 00 00 00 00
- 000 00 : 00 000 00000 000 (00) 00
- 00 00 : "0 000 XX 0000 00000 " 00 000 0 00
- 00 00 : 00 00 00 , 000 , 00 00 0 00 000 00
- 00 000 : LLM 000 00 000 0000 0

?? ??

- 00 000 00 : 000 0000 000 00
- 00 00 : 0000 000 0000 00 00
- 00 00 00 : 00 DB, 00 00 0 000 00
- 00 000 : 000 0000 0000 000

? ??? ?? ??

- 00 00 00 00 0
- 00 00000 00 00
- 00 00 000 (FAQ, 000)
- 00 /00 00 (00 00 00)
- 00 00 00
- 00 00 00 0 00

? MCP: LLM?? ??? ??

? MCP?? (Model Context Protocol)

LLM 模型 通过 调用 API 接口 获取 外部 数据 和 服务 (如 天气) 的能力。

? ? ??????

问题 1: LLM 模型 如何 调用 外部 API 接口 获取 数据 和 服务 的能力。

问题 2: LLM 模型 如何 调用 外部 API 接口 获取 数据 和 服务 的能力。

?? ?? ??

```
1. 用户: "帮我查询一下北京明天的天气"
   ↓
2. LLM 模型: "调用 API 接口 获取 天气" + "需要 天气 数据"
   ↓
3. MCP 代理: 调用 API 接口 → "明天 30 度 左右"
   ↓
4. MCP 代理: 返回 API 接口 → 返回 数据 "30 度 左右" 给 LLM
   ↓
5. LLM: 根据 返回 数据 生成 回答
   ↓
6. 用户: "明天 北京 天气 怎么样 30 度 左右 挺 好的"
```

? ??? ??

目前 支持 USB-C 接口的设备 包括 手机 和 笔记本电脑。

- MCP 代理: 通过 调用 API 接口 获取 数据 和 服务 (iPhone 的 Lightning, Android 的 USB-C)
- MCP 代理: 通过 AI 模型 生成 回答 和 服务 (如 天气 查询)

?? ??? ??

? ??

- 核心 : 构建 AI 应用 的 基础
- 数据 处理 : 清洗 数据 并 进行 分析
- 模型 训练 : API 调用 模型 训练 过程
- 模型 部署 : 将 训练 好的 模型 部署 到 生产 环境 (本地 部署 , 云 部署)
- 模型 评估 : 评估 模型 的 性能 和 准确性

?? ??

- 核心 概念 : 2024 年 的 核心 概念 和 技术
- 模型 训练 : MCP 模型 训练 的 流程和 方法
- 模型 部署 : MCP 模型 部署 的 API 接口 和 方法
- 模型 评估 : 如何 评估 模型 的 性能 和 准确性
- 模型 应用 : 将 模型 应用 到 实际 场景 中

? ??? ?? ??

- 模型 训练 (数据 , 模型 , 训练)
- 模型 部署 (Slack, Notion, Jira, GitHub)
- 模型 评估 (精度 , 召回 , 准确率)
- 模型 应用 (聊天 机器人 , 推荐 系统)
- IoT 模型 应用 (智能家居 , 工业 控制)
- 模型 训练 的 挑战 和 解决方案

? ?? ?? ?? ?? ??

核心 : 构建 AI 应用 的 基础

? MCP ??? (RAG?? ??)

- 核心 : "构建 AI 应用 的 基础"
- 模型 训练 : "模型 训练 的 流程和 方法"
- 模型 部署 : 将 模型 部署 到 生产 环境 , 本地 部署 和 云 部署

? MCP ???

1. 프롬프트: "리포트 작성해줘. 리포트는 어떤 내용이야?"

↓

2. LLM 프롬프트:

- "리포트" → Riot API 데이터 요청
- "리포트 작성해줘" → 리포트 API 호출

↓

3. MCP 프롬프트:

- Riot API 데이터 → "리포트 작성 52.3% (완료)"
- 리포트 API 호출 → "리포트 작성 50 30 20"

↓

4. LLM: 리포트 작성해줘

↓

5. 프롬프트: "리포트 작성해줘. 리포트는 어떤 내용이야?"

리포트는 2% 완료.

리포트 작성 50 30 20 (완료 60%)

리포트는 어떤 내용이야? 리포트

? ??? ??? MCP ??? ??

- 리포트 API 호출
- 리포트 API 호출
- 리포트 API 호출, 리포트 API 호출
- 리포트 API 호출 ("리포트 작성해줘" 리포트 작성해줘!)
- 리포트 API 호출 리포트 작성해줘

? MCP? ??? ??

리포트 "리포트 작성해줘" 리포트 작성해줘:

- 리포트 작성해줘 리포트 작성해줘 (리포트, 리포트, 리포트)
- 리포트 작성해줘 리포트
- 리포트 작성해줘 리포트 리포트
- 리포트 작성해줘 /리포트 리포트
- 리포트 작성해줘 리포트 리포트
- 리포트 작성해줘 리포트 (리포트, 리포트, 리포트)

리포트 리포트 리포트 리포트:

- "리포트 작성해줘 리포트 리포트"
- "리포트 작성해줘 리포트"

- "□□□ □□ □□ □□□ □□□ □"
- "□ □□ □□□ □□ □□□ □"

? RAG vs MCP: ??? ?????

? ?? ???

	RAG	MCP

? ?? ????? ??

"??? ?" ? ?

? ?? ??, ?? ??, ?? ? **RAG**

- "14.2 ? ? ?"
- "? ? ? ? ?"
- "? ? ? ? ?"

? ?? ??, ??? ??, ?? ??? ? **MCP**

- "? ? ? ?"
- "? ? ? ?"
- "? ? ? ? ?"

? ?? ?? ????? ?? ??

? RAG ???

- "S13 10 1000 1000 " → 10 10 10 10
- "1000 1000 1000 1000 ?" → 1000 1000 1000 1000
- "1000 1000 1000 " → 1000 1000 1000 1000

? MCP ???

- "10 10 10 1000 ?" → 1000 1000 API 1000
- "1000 1000 1000 1000 " → 1000 API + 1000 1000
- "1000 1000 1000 1000 1000 " → 1000 API 1000 + 1000 1000

? RAG + MCP ?? ??

1000 1000 1000 1000 1000

???: "?? ????? ????? ???? ? ?? ?? ?? ?? 5? ?????"

1. **RAG:** 10 10 10 1000 → 1000 1000 1000 1000
2. **LLM:** 10 10 1000 1000 (1000, 10, 1000, 1000, 1000)
3. **MCP:** Riot API 10 1000 1000 1000 1000
4. **LLM:** 1000 1000 1000 1000
5. 1000 :

"14.2 1000 1000 1000 1000 TOP 5:

1. 10 (54.2%) - Q 1000 1000 1000 1000
2. 1000 (52.3%) - 1000 1000 1000 1000
3. 1000 (51.8%) - 1000 1000 1000 1000
4. 1000 (51.2%) - W 1000 1000 1000 1000
5. 1000 (50.9%) - E 1000 1000 1000 1000

1000 1000 1000 1000 1000!"

? AI ??????: RAG + MCP? ??

? AI ????????

AI 에이전트 (AI Agent)는 LLM, RAG, MCP, 그리고 기타 기술을 결합하여 복잡한 작업을 수행할 수 있도록 설계된 소프트웨어 에이전트입니다.

LLM + RAG + MCP + 기타 기술은 AI 에이전트의 핵심 구성 요소입니다.

? AI ?????? ??

1. LLM: 텍스트 생성 및 이해를 위한 핵심 엔진
2. RAG: 외부 데이터베이스에서 관련 정보를 검색하여 LLM의 답변을 개선
3. MCP: 다양한 도구나 서비스에 연결하여 AI 에이전트의 기능을 확장
4. 기타: 사용자 인터페이스, 보안, 모니터링 등 에이전트의 운영을 위한 도구

? ? ?? AI ?????? ??????

이러한 에이전트는 LLM을 "두뇌"로 사용하고, RAG를 "메모리"로 활용합니다.

RAG와 MCP를 결합하면 LLM은 "두뇌" + "메모리" + "도구"를 갖춘 완전한 에이전트가 됩니다.

이러한 에이전트는 다양한 업무 프로세스를 자동화할 수 있습니다.

?? AI ?????? ?? ??



- API	
-	

-	
-	
-	

?? ?? ?? ??: ?? ?? ?? ??

?? : "?? ?? ?? ?? ?? ?? , ?? ?? ?? . ?? ?? ??
 ?? ?? ?? ?? ?? "

?? AI ????? ?? ??:

- ?? **LLM** ?? ??
 - ?? 1: ?? ?? ?? ?? ??
 - ?? 2: ?? ?? / ?? ??
 - ?? 3: ?? ?? ?? ??
- ?? **RAG** ?? ?? ??
 - ?? ?? DB ?? "?? ?? + ?? " ??
 - "14.3 ?? : Q ?? 4 → 3.5 , E ?? ?? " ??
 - ?? ?? ?? ?? / ?? ??
- ?? **MCP** ?? ?? ??
 - Riot API ?? : ?? ?? ?? 52.3%
 - ?? ?? / ?? ?? ??
 - ?? 1 ?? ?? ?? ??
- ?? **MCP** ?? ??
 - Discord Webhook ??
 - ?? ?? RSS ?? ?? ??
 - "?? " ?? ?? ?? ?? ?? ??
- ?? ?? ?? ?? ??

"14.3 ?? ?? ?? ?? ??!"

?? ?? ??:

- Q ?? : 4 → 3.5
- E ?? : +10 (?? ??)

?? ?? ?? (??+):

- 00: 52.3% (▲2.1%)

- 00: 8.7% (□□□□)

- 00: 12.4%

00 00: 000 000 S00 00 00

0 000 00 000 000000 0000000 0000000!

? ?? ???? ? ?? ??

? ?????? ?????

? ????? ???? ??????

□ □□ □□ □□ □□□□ :

? RAG ??

- □ □ □□ □□ ?
- □ □□ □□ □□ ?
- □ □ □□ □□ ?
- FAQ □ □□ □□ ?

? MCP ??

- □□ □ □□ □□ ?
- □ □□ □□ □□ ?
- □□ □ □□ □□ □□ ?
- □ □□ □□ □□ ?

? ??? ?? ??

1?? ?? LLM ??

- □□ □□ □□□□
- □□□□ □□ □□

2?? RAG ??

- □□ □□ , □□ □□
- □□□ □□ □□

3?? MCP ??

- 1-2
-

4?? AI ?????

- RAG + MCP
-
-

? ?? ???? ??????

? RAG? ????? ? ?

" " :

-
- ,
-
-

- "14.3 ??"
- " 5 ?" "
- " ??"

? MCP? ????? ? ?

" " :

- (, ,)
-
-
- /
-

□□ □□

- "□□□ □□ □□ □□□ □"
- "□ □ □□ □□□ □"
- "□□□ □□ □□ □□□ □□□ □"
- "□ □□ □□□ □□ □□□ □"

? RAG + MCP? ?? ??? ?

□□□□ □□□ □□□ □ <:

□□ : "□□ □□□□ □□□□ □□□ □ □□ □□ □□ □□ 5□ □□□ □"

1. **RAG**: □□ □□ □□ □□ → □□□□ □□□ □□
2. **MCP**: API□ □ □□□□ □□ □□ □□
3. **LLM**: □□□ □□ □ □□ □□

? ?? ??

? ? ?? ??

- **LLM** □□: □□□ □ □□ □□□ AI
- **RAG** □□: LLM□□ □□ □□□ □□ □□
- **MCP** □□: LLM□□ □□□ □□□□ □□
- **AI** □□□□ □□: RAG + MCP□ □□ □□ □□□□ AI □□

? ????? ? ??

1. LLM? ?????? ??? ? ? ? ?

- □□ □□□ □□ □□□ □□
- □□ □□ □□ □□ □□
- □□ □□□ □□ □□

2. RAG? "?? ?" ? ? ? ? ?

- □□ □□□□ □□ □□ □□
- □□□ □□ □□
- □□ □□ □□

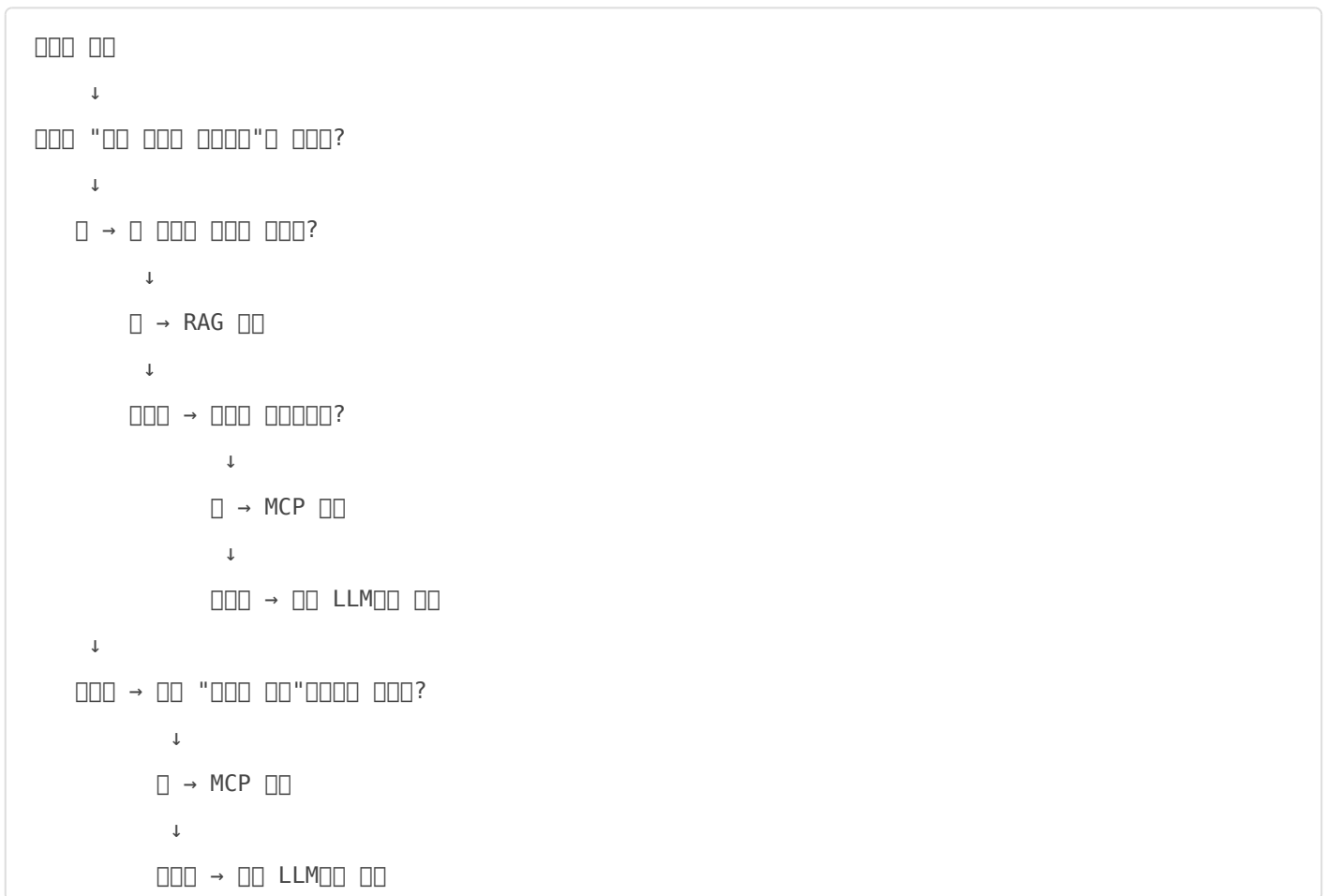
3. MCP? "? ? ? ? ?" ? ? ? ? ? ?

- 1 1 1
- 1 1 1 1
- 1 1 1

4. AI ????? ? ?? ? ? ? ? ? ?

- 1 (LLM)
- 1 (RAG)
- 1 (MCP)
- 1 1

?? ?? ????? ??????



? ?? ?

? ??? ??

1. LLM
- 2.
3. RAG
4. MCP

??????

-
-
-

??????

-
-
-
-

????

LLM, RAG, MCP

- **LLM**
- **RAG**
- **MCP**

, AI

(RAG), (MCP), (LLM)

!