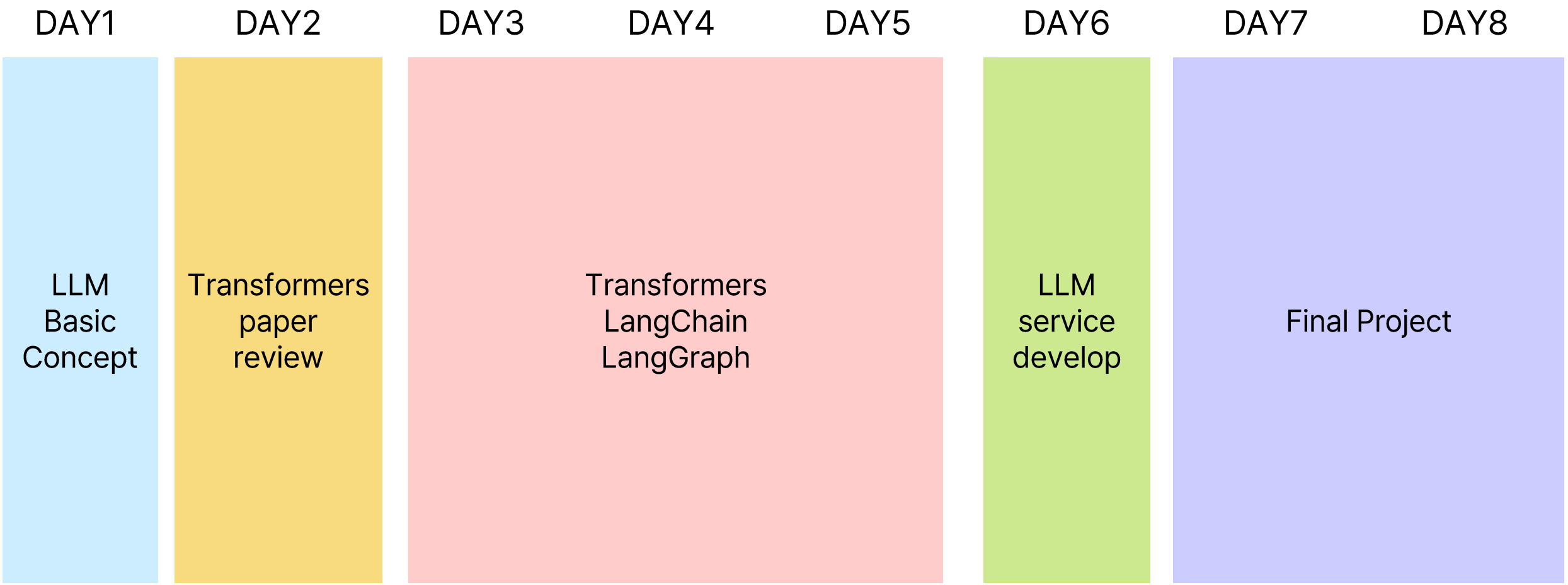
GEN AI 인텐시브 과정

강사 장철원

Section 0

코스소개



Section 0

목차

❑ 트랜스포머 + RAG 실습

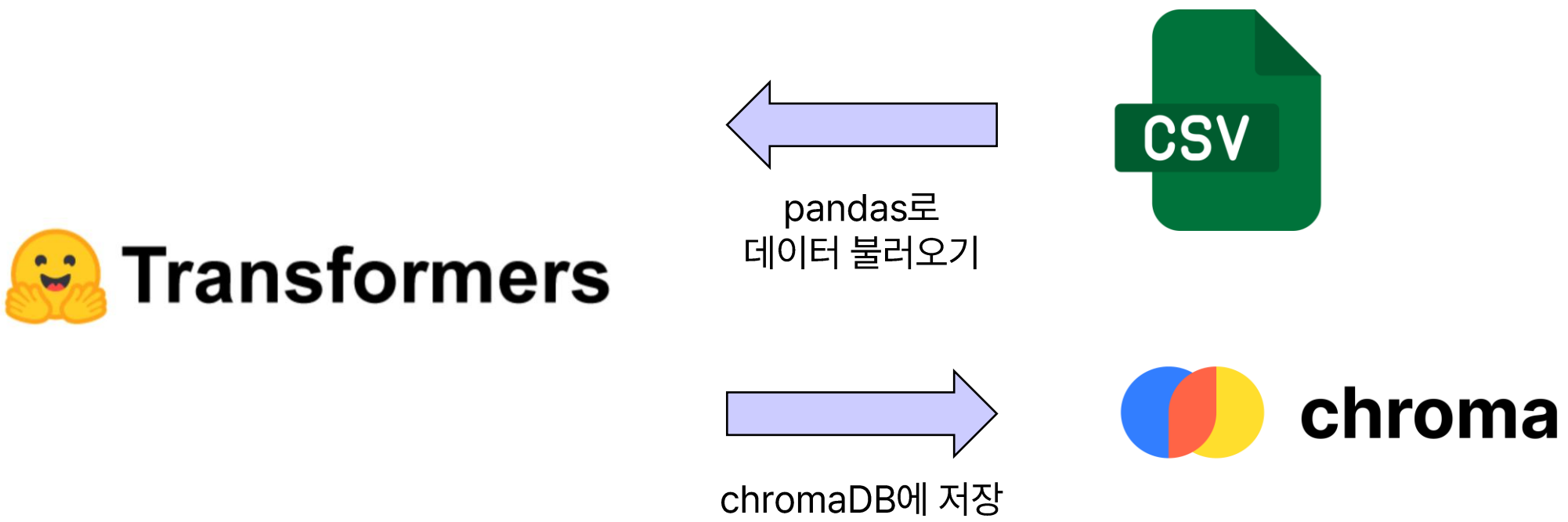
❑ 트랜스포머 + RAG + chromaDB 실습

GEN AI 인텐시브 과정

Section 2. RAG 실습

Section 2-1. 트랜스포머 + RAG

실습 내용



Section

트랜스포머+RAG

데이터 불러오기

```
import pandas as pd
import torch
import chromadb
from sentence_transformers import SentenceTransformer
from transformers import AutoTokenizer, AutoModelForQuestionAnswering
```

```
# 1. 사내 문서 로딩
df = pd.read_csv('./data/data.csv', encoding='utf8')
df.head(10)
```

	text
0	Etching 공정 전에는 반드시 세정 공정이 완료되어야 합니다.
1	PM 장비의 필터는 주 1회 정기적으로 교체해야 합니다.
2	웨이퍼 투입 전 챔버 내부의 온도 안정화가 필요합니다.
3	불량률이 2%를 초과할 경우 원인 분석 보고서를 제출해야 합니다.
4	클린룸 입장 전에는 반드시 정전기 방지복을 착용해야 합니다.
5	포토 공정 시 PR 코팅 두께는 1.5μm 이상 유지해야 합니다.
6	검사 장비는 매일 초기화 후 기능 점검을 실시합니다.
7	주간 생산 계획은 매주 월요일 오전 9시에 확정됩니다.
8	X선 검사 장비는 비정상 신호 발생 시 즉시 사용 중지합니다.
9	로더 설비의 진공 펌프는 월 1회 이상 윤활 상태를 점검합니다.

Section
트랜스포머+RAG

리스트 형태로 바꾸기

```
sc_list = df['text'].tolist()
sc_list
```

```
['Etching 공정 전에는 반드시 세정 공정이 완료되어야 합니다.',
 'PM 장비의 필터는 주 1회 정기적으로 교체해야 합니다.',
 '웨이퍼 투입 전 챔버 내부의 온도 안정화가 필요합니다.',
 '불량률이 2%를 초과할 경우 원인 분석 보고서를 제출해야 합니다.',
 '클린룸 입장 전에는 반드시 정전기 방지복을 착용해야 합니다.',
 '포토 공정 시 PR 코팅 두께는 1.5μm 이상 유지해야 합니다.',
 '검사 장비는 매일 초기화 후 기능 점검을 실시합니다.',
 '주간 생산 계획은 매주 월요일 오전 9시에 확정됩니다.',
 'X선 검사 장비는 비정상 신호 발생 시 즉시 사용 중지합니다.',
 '로더 설비의 진공 펌프는 월 1회 이상 윤활 상태를 점검합니다.',
 '공정 이탈 발생 시 Shift 리더에게 즉시 보고합니다.',
 '동일 Lot 내에서 불량이 5개 이상 발생하면 전수 검사 실시합니다,',
 '물류 이송 로봇은 경로 장애 감지 시 자동 우회합니다.',
 '이온 주입 공정 후 열처리 시간은 최소 30분 이상 확보합니다.',
 '클린룸 청소는 일 2회, 장비 청소는 주 1회 이상 실시합니다.',
 '신입 엔지니어는 공정별 교육을 모두 이수한 후 장비 조작이 가능합니다.',
 '공정 레시피 변경 시 Change History 문서 작성을 필수로 합니다.',
 '수율 개선안은 월간 품질 회의에서 발표되어야 합니다.',
 'Mask alignment 오류가 0.2μm를 초과하면 장비 점검을 실시합니다.',
 'MES 시스템에 기록된 공정 로그는 6개월간 보관됩니다.',
 '웨이퍼 저장 캐리어는 주기적으로 오염도를 측정해야 합니다.',
 '장비 재가동 시 Warm-up 절차를 반드시 이행합니다.',
 'Test wafer는 생산 wafer 투입 전에 반드시 시뮬레이션합니다.',
 '제조 Lot ID는 공정마다 바코드로 자동 추적됩니다.',
 '소자 특성 분석 결과는 전자문서 시스템에 등록합니다.',
```

문장 임베딩

```
# 2. 문장 임베딩
embedding_model = SentenceTransformer("sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2")
```

```
ids_list = []
ebd_list = []
text_list = []

for i, sc in enumerate(sc_list):
    doc_id = f"doc{i+1}"
    ids_list.append(doc_id)
    text_list.append({"text": sc})

    embedding = embedding_model.encode(sc).tolist()
    ebd_list.append(embedding)
```

Section

트랜스포머 + RAG

ids_list

```
['doc1',
 'doc2',
 'doc3',
 'doc4',
 'doc5',
 'doc6',
 'doc7',
 'doc8',
 'doc9',
 'doc10',
 'doc11',
 'doc12',
 'doc13',
 'doc14',
 'doc15',
 'doc16',
 'doc17',
 'doc18',
 'doc19',
 'doc20',
 'doc21',
 'doc22',
 'doc23',
 'doc24',
 'doc25',
 'doc26',
 'doc27',
 'doc28',
 'doc29',
 'doc30']
```

ebd_list

```
[[-0.04117877408862114,
  0.07787931710481644,
  0.04467811807990074,
  -0.02135879546403885,
  -0.20158037543296814,
  -0.020650850608944893,
  -0.3815535008907318,
  0.18558235466480255,
  -0.16642484068870544,
  0.10929572582244873,
  -0.08098505437374115,
  0.13745461404323578,
  0.16525739431381226,
  -0.16476202011108398,
  -0.09935851395130157,
  -0.08239393681287766,
  6.90582673996687e-05,
  -0.13697274029254913,
  -0.11696404963731766,
  -0.1441047340631485,
  0.13622133433818817,
  -0.13961464166641235,
  -0.014541568234562874,
  -0.1833028346300125,
  0.1281987577676773,
  0.08415351063013077,
  -0.098037488758564,
  -0.11522170901298523,
  0.20508992671966553,
  -0.14426401257514954,
  -0.08760103583335876,
```

text_list

```
[{'text': 'Etching 공정 전에는 반드시 세정 공정이 완료되어야 합니다.'},
 {'text': 'PM 장비의 필터는 주 1회 정기적으로 교체해야 합니다.'},
 {'text': '웨이퍼 투입 전 챔버 내부의 온도 안정화가 필요합니다.'},
 {'text': '불량률이 2%를 초과할 경우 원인 분석 보고서를 제출해야 합니다.'},
 {'text': '클린룸 입장 전에는 반드시 정전기 방지복을 착용해야 합니다.'},
 {'text': '포토 공정 시 PR 코팅 두께는 1.5μm 이상 유지해야 합니다.'},
 {'text': '검사 장비는 매일 초기화 후 기능 점검을 실시합니다.'},
 {'text': '주간 생산 계획은 매주 월요일 오전 9시에 확정됩니다.'},
 {'text': 'X선 검사 장비는 비정상 신호 발생 시 즉시 사용 중지합니다.'},
 {'text': '로더 설비의 진공 펌프는 월 1회 이상 윤활 상태를 점검합니다.'},
 {'text': '공정 이탈 발생 시 Shift 리더에게 즉시 보고합니다.'},
 {'text': '동일 Lot 내에서 불량이 5개 이상 발생하면 전수 검사 실시합니다,'},
 {'text': '물류 이송 로봇은 경로 장애 감지 시 자동 우회합니다.'},
 {'text': '이온 주입 공정 후 열처리 시간은 최소 30분 이상 확보합니다.'},
 {'text': '클린룸 청소는 일 2회, 장비 청소는 주 1회 이상 실시합니다.'},
 {'text': '신입 엔지니어는 공정별 교육을 모두 이수한 후 장비 조작이 가능합니다.'
 {'text': '공정 레시피 변경 시 Change History 문서 작성을 필수로 합니다.'},
 {'text': '수율 개선안은 월간 품질 회의에서 발표되어야 합니다.'},
 {'text': 'Mask alignment 오류가 0.2μm를 초과하면 장비 점검을 실시합니다.'},
 {'text': 'MES 시스템에 기록된 공정 로그는 6개월간 보관됩니다.'},
 {'text': '웨이퍼 저장 캐리어는 주기적으로 오염도를 측정해야 합니다.'},
 {'text': '장비 재가동 시 Warm-up 절차를 반드시 이행합니다.'},
 {'text': 'Test wafer는 생산 wafer 투입 전에 반드시 시뮬레이션합니다.'},
 {'text': '제조 Lot ID는 공정마다 바코드로 자동 추적됩니다.'},
 {'text': '소자 특성 분석 결과는 전자문서 시스템에 등록합니다.'},
 {'text': '야간조 작업자는 작업 시작 전 점검 체크리스트를 반드시 확인합니다.'},
 {'text': '공정 조건이 기준에서 벗어난 경우 자동 알람이 발생합니다.'},
 {'text': '생산 중단 요청은 생산 관리팀 승인 후 진행할 수 있습니다.'},
 {'text': '사내 기술 포럼 자료는 R&D 인트라넷에 주기적으로 업데이트됩니다.'},
 {'text': '정전 발생 시 UPS 시스템으로 30분간 운영이 가능합니다.'}]
```

chromaDB 저장

```
# 3. chromaDB에 저장
client = chromadb.PersistentClient(path="./chromaDB")
collection = client.get_or_create_collection("company_doc")

collection.add(
    ids=ids_list,
    embeddings=ebd_list,
    metadatas=text_list
)
```

Section

트랜스포머+RAG

질문 임베딩

```python
# 4. 사용자 질문 입력 & 임베딩
question = "MES 공정 로그의 보관 기간은?"
question_embedding = [embedding_model.encode(question).tolist()]
print(question_embedding)
```

```
[[-0.09548699110746384, 0.18630450963974, -0.08783216029405594, -0.15297633409500122, 0.06896570324897766, -0.186129942
0.05591766536235809, -0.0965639054775238, 0.21273842453956604, 0.012247104197740555, -0.054250217974185944, 0.243348747
24307450652122498, -0.10421781986951828, 0.08761274069547653, -0.3318907916545868, 0.16946543753147125, -0.021498398855
715633124113083, -0.17815536260604858, -0.09230948239564896, 0.012314221821725368, 0.08089997619390488, -0.144365295767
6174428854137659, -0.1414806842803955, 0.08459586650133133, -0.07305973023176193, 0.16202040016651154, -0.07475023716688
9699685573578, 0.21895694732666016, -0.32925164699554443, -0.129502072930336, -0.14442576467990875, -0.1529991626739502,
307444572, 0.2769106924533844, 0.03755395486950874, 0.21262450516223907, 0.09466126561164856, -0.2017543911933899, -0.49
04, 0.24254445731639862, 0.022113284096121788, 0.03799907863140106, 0.16716811060905457, -0.06125044450163841, 0.288218
0.20166930556297302, -0.013401069678366184, 0.13967083394527435, 0.09382151812314987, -0.0818871334195137, -0.379878520
03889474645256996, 0.10407095402479172, 0.008736283518373966, -0.280147910118103, 0.32033661007881165, -0.12871004641056
391574382782, -0.2629503011703491, 0.05180894210934639, -0.030377190560102463, 0.19168877601623535, 0.23949456214904785,
6233368, -0.08391029387712479, 0.16241592168807983, -0.25032445788383484, 0.17458859086036682, -0.2469562441110611, -0.1
523792, -0.037469372153282166, -0.02234671637415886, -0.07063358277082443, -0.3305300772190094, 0.12567496299743652, -0.
150879, 0.12946981191635132, 0.0229450985789299, -0.04074897617101669, -0.2751513719558716, 0.30653250217437744, 0.24834
0.05132363736629486, 0.17061011493206024, -0.3940853178501129, -0.07183846831321716, -0.16599343717098236, 0.1337330788
48640090227127, -0.13282383978366852, 0.012419234961271286, -0.01448604092001915, 0.043774351477622986, 0.04365256056189
7984142303467, 0.16883231699466705, -0.032258786261081696, 0.09227271378040314, 0.17103776335716248, -0.251555681228637
4622324705124, 0.18086351454257965, -0.3083222508430481, -0.15130464732646942, -0.093690425157547, 0.15750330686569214,
3233966827, -0.23401698470115662, -0.006974874064326286, -0.05718585103750229, -0.030344484373927116, 0.1836023181676864
```

가장 유사한 문서 검색

```
# 5. 가장 유사한 문서 검색
result = collection.query(query_embeddings=question_embedding, n_results=1)
retrieved_text = result['metadatas'][0][0]['text']
print(f"선택된 문서: {retrieved_text}")
```

```
선택된 문서: MES 시스템에 기록된 공정 로그는 6개월간 보관됩니다.
```

Section
트랜스포머+RAG

모델 불러오기 & 답변 출력

```
# 6. QA 모델 로딩
qa_model_name = "monologg/koelectra-base-v3-finetuned-korquad"
qa_tokenizer = AutoTokenizer.from_pretrained(qa_model_name)
qa_model = AutoModelForQuestionAnswering.from_pretrained(qa_model_name)
```

```
# 7. QA 모델에 질문 + 문서 입력
inputs = qa_tokenizer(question, retrieved_text, return_tensors="pt", truncation=True, padding=True, max_length=512)

with torch.no_grad():
    outputs = qa_model(**inputs)

start_index = torch.argmax(outputs.start_logits)
end_index = torch.argmax(outputs.end_logits) + 1

answer = qa_tokenizer.decode(inputs["input_ids"][0][start_index:end_index], skip_special_tokens=True)
```

```
# ===== 8. 출력 =====
print(f"질문: {question}")
print(f"정답: {answer}")
```

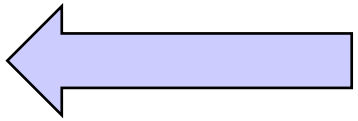
```
질문: MES 공정 로그의 보관 기간은?
정답: 6개월
```

GEN AI 인텐시브 과정

Section 2. RAG 실습

Section 2-2. 트랜스포머 + RAG + from chromaDB 실습

실습 내용



chromaDB에서
데이터 불러오기

Section

트랜스포머 + RAG + chromaDB

트랜스포머

```
import torch
import chromadb
from sentence_transformers import SentenceTransformer
from transformers import AutoTokenizer, AutoModelForQuestionAnswering
```

```
# 1. 임베딩 모델 로딩
embedding_model = SentenceTransformer("sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2")
```

```
# 2. chromaDB 불러오기
client = chromadb.PersistentClient(path="./chromaDB")
collection = client.get_or_create_collection("company_doc")
```

```
# 3. 사용자 질문 & 임베딩
question = "MES 공정 로그의 보관 기간은?"
question_embedding = [embedding_model.encode(question).tolist()]
```

```
# 4. 유사한 문서 검색
result = collection.query(query_embeddings=question_embedding, n_results=1)
retrieved_text = result['metadatas'][0][0]['text']
print(f"선택된 문서: {retrieved_text}")
```

```
선택된 문서: MES 시스템에 기록된 공정 로그는 6개월간 보관됩니다.
```

Section

트랜스포머 + RAG + chromaDB

트랜스포머

```python
# 5. QA 모델 로딩
qa_model_name = "monologg/koelectra-base-v3-finetuned-korquad"
qa_tokenizer = AutoTokenizer.from_pretrained(qa_model_name)
qa_model = AutoModelForQuestionAnswering.from_pretrained(qa_model_name)
```

```python
# 6. QA 모델 입력 & 실행
inputs = qa_tokenizer(
    question,
    retrieved_text,
    return_tensors="pt",
    truncation=True,
    padding=True,
    max_length=512
)
```

```python
# 7. 결과 출력
with torch.no_grad():
    outputs = qa_model(**inputs)

start_index = torch.argmax(outputs.start_logits)
end_index = torch.argmax(outputs.end_logits) + 1
answer = qa_tokenizer.decode(inputs["input_ids"][0][start_index:end_index], skip_special_tokens=True)

print(f"질문: {question}")
print(f"정답: {answer}")
```

```
질문: MES 공정 로그의 보관 기간은?
정답: 6개월
```

감사합니다.

Q & A