

책을 출판하려고 한다. 책의 뒷부분의 인덱스를 뽑으려고 하는데,
일반적으로 인덱스는 어떤 것이 좋을까?

책의 인덱스(Index)는 독자가 특정 내용을 쉽게 찾을 수 있도록 도와주는 중요한 요소입니다. 보통 다음과 같은 기준을 고려하여 인덱스를 작성합니다.

1. 주요 개념 및 용어

- 책에서 자주 등장하거나 핵심 개념인 용어
- 예: 딥러닝, 자연어 처리, RAG, 파인 투닝, 임베딩

2. 중요한 인물 및 조직

- 특정 연구자, 개발자, 기업, 기관 등
- 예: OpenAI, LangChain, Yann LeCun, Google DeepMind

3. 기술 및 알고리즘

- 설명한 알고리즘, 프레임워크, 프로그래밍 언어 등
- 예: Transformer, GPT-4o, CLIP, PyTorch, LangChain

4. 코드 및 명령어

- 독자가 특정 코드 예제를 쉽게 찾을 수 있도록 코드 관련 용어 포함
- 예: cosine_similarity, SentenceTransformer, fine-tuning

5. 실제 예제 및 사례

- 책에서 다룬 주요 사례나 응용 프로그램
- 예: RAG를 활용한 문서 검색, AI를 활용한 챗봇 구축

6. 약어 및 키워드

- AI 분야에서 자주 쓰이는 약어나 키워드
- 예: LLM (Large Language Model), API, NLP

7. 각 장별 주요 주제

- 각 장에서 강조하는 주요 개념을 인덱스에 포함
- 예: Chapter 7 - AI 챗봇 구축, Chapter 10 - OpenAI API 활용법

8. 관련 개념 연결

- 유사하거나 연관된 개념을 서로 연결
- 예: 임베딩 → 벡터 검색, 클러스터링 → K-means

9. 한글 및 영어 병기 (필요할 경우)

- 독자가 검색할 때 혼동되지 않도록 중요한 용어를 한글과 영어로 병기
- 예: 자연어 처리 (Natural Language Processing, NLP)

인덱스 생성 시, 각 키워드에 페이지 번호를 포함해야 하며, 자동화된 방법(예: LaTeX, MS Word의 색인 기능)을 활용하면 효율적입니다.

어떤 방식으로 만들지 고민 중이라면, 책의 주제와 독자층을 고려하여 중요한 개념 위주로 선정하는 것 이 좋습니다. 필요하면 샘플 인덱스를 만들어볼 수도 있습니다! 😊

책을 출판하려고 한다. 책의 뒷부분의 인덱스를 뽑으려고 하는데,
일반적으로 인덱스는 어떤 것이 좋을까?

4초 동안 인덱스 작성에 대해 고민하셨다면, 책의 주요 개념과 용어, 그리고 독자가 중요한 내용에 쉽게 접근할 수 있도록 만드는 것이 중요합니다.>

책의 인덱스는 독자가 원하는 정보를 빠르게 찾을 수 있도록 도와주는 중요한 도구입니다. 좋은 인덱스를 구성하기 위한 일반적인 요소와 방법은 다음과 같습니다.

1. 핵심 용어 및 주제 선정

- **주요 개념 및 용어:** 책에서 다루는 핵심 개념, 용어, 이론 등을 포함합니다.
예) "자연어 처리", "딥러닝", "빅데이터"
- **관련 하위 항목:** 주요 용어에 대한 세부 항목이나 연관 개념을 함께 나열합니다.
예) "자연어 처리 → 텍스트 분석", "딥러닝 → 신경망 구조"

2. 인물, 조직 및 사례

- **주요 인물 및 기관:** 책에서 언급된 인물, 조직, 회사 등도 포함시켜 독자가 관련 정보를 쉽게 찾을 수 있도록 합니다.
예) "스티븐 호킹", "구글"
- **실제 사례 및 사례 연구:** 특정 사례나 연구 내용이 중요한 경우, 이를 인덱스에 포함시켜 필요할 때 바로 찾아볼 수 있도록 합니다.

3. 기술적 요소 및 방법론

- **알고리즘 및 기술:** 기술 서적의 경우, 각종 알고리즘, 프로그래밍 언어, 기술 용어 등을 정리합니다.
예) "알고리즘 → 정렬", "프로그래밍 언어 → Python"
- **코드 예제 및 명령어:** 코드나 스크립트가 중요한 경우, 해당 코드의 주요 부분이나 함수, 라이브러리 이름 등을 포함합니다.

4. 체계적 구성 및 표기법

- **일관된 표기:** 한글과 영어 용어가 혼용될 경우, 일관되게 표기(예: "자연어 처리 (NLP)")하여 독자의 혼란을 줄입니다.
- **계층적 구성:** 주요 항목과 하위 항목을 계층적으로 정리하여, 보다 세밀하게 분류할 수 있도록 합니다.
- **페이지 번호 기재:** 각 항목에 해당하는 페이지 번호를 정확하게 기재해 독자가 쉽게 해당 내용을 찾아볼 수 있도록 합니다.

5. 자동화 도구 활용

- **인덱스 자동 생성 기능:** 워드 프로세서나 LaTeX 등의 도구에서 제공하는 자동 인덱스 기능을 활용하면, 수정이나 업데이트 시 효율적입니다.
- **검토 및 수정:** 초안 작성 후, 인덱스 항목을 재검토하고 독자의 입장에서 빠진 내용이나 과도한 항목이 없는지 점검하는 것이 좋습니다.

이와 같이 독자의 편의성을 고려하고, 책의 주제와 구성에 맞춰 핵심적인 내용들을 정리하면 좋은 인덱스를 만들 수 있습니다.

책은 제목은 아래와 같고, 다른은 목차는 아래와 같다.

[제목] 제대로 시작하는 챗GPT와 AI 활용 with 파이썬

[목차]

PART 1 오픈AI와 챗GPT 제대로 이해하기

1장 생성형 AI 서비스

1.1 생성형 AI 개요

..... 25

1.1.1 생성형 AI의 활용

..... 25

1.1.2 생성형 AI 노션의 아키텍처

..... 26

1.2 대표적인 생성형 AI 서비스	26
2장 오픈AI와 챗GPT	
2.1 오픈AI와 챗GPT 개요	29
2.2 GPT 모델 및 사용 사례	33
3장 개발 사전 준비	
3.1 구글 코랩 사용 방법	37
3.2 API 개요	40
3.3 오픈AI API 키 얻기	41
3.4 오픈AI API 키 활용 실습	45
3.4.1 openai 라이브러리 설치	45
3.4.2 chatgpt.env 파일 작성	46
3.5 API 사용량 및 요금 확인	56
4장 오픈AI API 활용 첫걸음	14
4.1 GPT-4o mini로 시작하는 대화 생성	59
4.2 system, user, assistant 역할의 기본 이해	63
4.2.1 system 역할	63
4.2.2 user 역할	65
4.2.3 assistant 역할	66
4.3 퓨샷 학습 소개	66
4.4 GPT 모델을 활용한 맞춤형 출력 생성	68
4.5 대화 출력 길이 제어	72
4.6 AI 응답 제어	72
↓	76
5장 오픈AI API 모델 탐색하기	
5.1 사용 가능 모델 확인 및 소개	81
5.1.1 GPT 시리즈: 텍스트 생성 및 이해	85
5.1.2 DALL-E 시리즈: 이미지 생성	85
5.1.3 워스퍼 시리즈: 음성 인식	85
5.1.4 TTS: 텍스트의 음성 변환	85
5.1.5 텍스트 임베딩 모델	86
5.2 사용할 모델 선택	86
5.3 오픈AI 모델 시리즈	87
5.3.1 GPT-4o 시리즈	87
5.3.2 GPT-4o mini 시리즈	88
5.3.3 o1 및 o1-mini 시리즈	89
5.3.4 GPT-4 Turbo 및 GPT-4 시리즈	90
5.3.5 DALL-E 시리즈	

.....	90
5.3.6 TTS 시리즈	
.....	91
목차	
15	
5.3.7 워스퍼 시리즈	
.....	91
5.3.8 임베딩 모델	
.....	92
5.4 오픈AI 모델 가격	
.....	92
5.4.1 GPT-4o 모델 가격	
.....	93
5.4.2 GPT-4o mini	
.....	94
5.4.3 o1 및 o1-mini 가격	
.....	95
5.4.4 임베딩 모델 ↓	
.....	95
5.4.5 파인 투닝 모델	
.....	96
5.4.6 실시간 API	
.....	97
5.4.7 어시스턴트 API	
.....	98
5.4.8 이미지 모델	
.....	98
5.4.9 오디오 모델	
.....	99
5.4.10 기타 모델	
.....	99
5.5 오픈AI 모델의 발전과 선택 가이드	
.....	100
PART 2 오픈AI GPT 대화 생성 실전 가이드	
6장 대화 생성의 고급 기술	
6.1 창의성 조절 기법	
6.1.1 temperature	
.....	104
6.1.2 top_p 샘플링	
.....	107
6.1.3 temperature와 top_p의 차이점	
.....	110
6.2 페널티로 독창성과 다양성 확보하기	
.....	111
6.3 n 파라미터로 여러 응답 생성하기	
.....	115
6.4 실시간 스트리밍으로 즉시 출력하기	
.....	117
16	
7장 GPT 모델의 성능을 극대화하는 프롬프트 엔지니어링 기법	
7.1 프롬프트 엔지니어링 개요	
.....	119
7.2 퓨샷 학습: 키워드 추출 성능 높이기	
.....	121
7.3 프롬프트 체이닝: 트윗 생성 최적화하기	
.....	127
7.4 일반 지식 프롬프팅: 올드 스쿨 랩 가사 완성하기	
.....	133
7.5 문맥 채우기: AI의 이해도 높이기	
.....	138
7.6 동적 토큰 제어: AI 응답 길이 유연하게 조절하기	
.....	147
7.7 프롬프트 템플릿 최적화: 리눅스 학습용 CLI 어시스턴트 개발	
하기	153
7.8 다음으로 나아가기	
.....	164
PART 3 음성 및 이미지 인식·의스퍼와 CLI의 신저 활용	

8장 위스퍼 음성 인식 기초	
8.1 위스퍼 개요	169
8.2 위스퍼 설치 방법 및 기본 예제	171
8.3 파이썬에서 위스퍼 SDK 사용하기	173
9장 위스퍼 텍스트 변환 최적화 기법	
9.1 오픈AI 음성-텍스트 API 사용하기	177
9.1.1 전사 API	177
9.1.2 번역 API	179
9.2 위스퍼 텍스트 변환 개선하기	180
9.2.1 오디오 정리하기	180
9.2.2 프롬프트 파라미터 사용하기	181
목차	17
9.2.3 전사된 텍스트 후처리하기	183
10장 오픈AI TTS를 활용한 음성 변환	
10.1 오픈 AI TTS 소개	185
10.2 오픈 AI TTS 사용 방법	186
11장 오픈AI CLIP을 사용한 이미지 분류	
11.1 CLIP 소개	191
11.2 CLIP 사용 방법	191
11.2.1 필수 라이브러리 설치	192
11.2.2 CLIP으로 ↓로션 분류하기	192
11.3 이미지를 텍스트로 변환하기	199
PART 4 생성형 AI를 활용한 이미지 제작과 편집 기술	
12장 DALL-E로 이미지 생성하기	
12.1 DALL-E 소개	203
12.2 프롬프트 기반의 이미지 생성 기본 예제	205
12.3 다른 크기의 이미지 생성하기	207
12.4 여러 이미지 생성하기	208
12.5 이미지 프롬프트 개선 방법	210
12.5.1 작가 모방	210
12.5.2 예술 스타일 모방	213
12.5.3 분위기와 감정	215
12.5.4 색상	219
12.5.5 해상도	221
12.5.6 각도와 위치	221
12.5.7 렌즈 종류와 촬영 기법	223

12.5.9 이미지 필터

12.6 프롬프트 조합으로 다양한 랜덤 이미지 만들기

13장 DALL-E로 이미지 편집하기

13.1 이미지 편집 예제

14장 다른 이미지에서 영감 얻기

14.1 이미지 변형 방법

14.2 이미지 변환 용 사례

PART 5 임베딩: 복잡한 데이터를 쉽게 이해하는 방법

15장 임베딩 소개

15.1 임베딩의 의미

15.2 임베딩 사용 사례

15.2.1 테슬라: 자율주행차 학습 효율성 향상

15.2.2 Kalendar AI: 세일즈 타겟팅 정확도 향상

15.2.3 노션: 검색 기능 향상

15.2.4 DALL-E 3: 텍스트의 이미지 변환 능력 향상

목차

16장 텍스트 임베딩 활용 방법

16.1 텍스트 임베딩의 핵심 이해하기

16.2 다중 입력 사용 예제

16.3 사용자의 의도를 읽어내는 시맨틱 검색

16.4 코사인 유사도 쉽게 이해하기

16.5 오픈AI 텍스트 임베딩 사용 예제

16.6 임베딩의 내부 작동 원리

17장 고급 임베딩 예제

17.1 선호하는 커피 예측하기

17.2 퍼지 검색으로 더 유연한 커피 검색 구현하기

17.3 제로샷 분류기로 뉴스 기사 분류하기

17.4 제로샷 분류기의 정밀도 평가

17.5 제로샷 분류기의 정밀도 활용 예시

PART 6 파인 튜닝과 모델의 실전 활용

18장 퓨샷 학습 이해하기

18.1 퓨샷 학습의 개념과 정의

18.2 파인 튜닝으로 퓨샷 학습 성능 향상하기

19장 파인 튜닝의 이해 및 구현

19.1 파인 튜닝 실제 적용

19.2 파인 튜닝 최적화

19.2.1 모델 선택

19.2.2 데이터셋 검증	331
19.2.3 토큰 제한	340
19.2.4 데이터셋 크기	340
19.2.5 훈련 테스트 및 개선: 하이퍼파라미터 조정	340
19.2.6 예상 비용 고려하기	344
19.2.7 데이터셋 품질 높이기	345
19.2.8 파인 투닝과 다른 기술 결합하기	346
19.2.9 실험과 학습 지속하기	346
19.2.10 모델 테스트하기	347
19.2.11 검증 세트 사용하기	347
19.2.12 결과 분석하기	348
20장 파인 투닝 고급 예제: 정신 건강 코치	
20.1 예제에서 사용한 데이터셋	353
20.2 데이터셋 준비하기	355
20.3 실제 애플리케이션에서 발생할 수 있는 한계	366
21장 기억력 및 문맥 문제와 해결책	
21.1 GPT의 기억력 및 문맥 문제	369
21.2 문맥 유지의 중요성	370
21.3 대화 문맥 유지의 해법: 대화 이력 추가	373
21.4 대화 이력 관리의 문제점	377
21.5 후입선출 방식으로 최신 문맥 유지하기	378
21.6 후입선출 메모리의 한계	385
21.7 선택적 문맥으로 최적의 대화 문맥 선택하기	385
목차	
21	
PART 7 부록	
부록 A 챗GPT 및 클로드 실습	
A.1 코딩 공부에 활용하기	399
A.2 유튜브 영상에서 텍스트를 추출하고 요약하기	405
A.3 간단한 고객 데이터 분석 수행하기	411
A.4 맞춤형 GPT 소개 및 구현	414
부록 B 오픈AI 플레이그라운드 실습	
B.1 오픈AI 플레이그라운드 소개	429
B.2 취업 진로 상담하기	434
B.3 이모지 만들기	437
B.4 웹 페이지 작성하기	441
부록 C 건강 상담 챗봇 구현하기	
C.1 스트림릿으로 간단한 웹 앱 서비스 구현하기	445

C.2 건강 상담 챗봇 웹 서비스 구현	452
부록 D 작가 스타일 이미지 생성 앱	
D.1 풍경화 이미지 생성하기	
부록 E RAG로 노트북 추천받기	460
E.1 RAG 소개	
부록 F 오토젠플로팅 활용한 AI 에이전트 이해하기	471
F.1 오토젠플로팅 소개	
F.2 멀티 에이전트 활용 실습	485
찾아보기	486
	501

위의 내용을 토대로 index를 만들어볼수 있니?

20초 동안 이성 처리 완료 >

아래는 목차 내용을 참고하여 구성한 인덱스의 예시입니다. 실제 출판 시에는 페이지 번호가 최종 편집본에 맞게 수정되어야 하지만, 아래 예시는 각 항목별로 독자가 쉽게 관련 내용을 찾아볼 수 있도록 구성한 샘플입니다.

예시 인덱스

API

- API 개요, 40
- 오픈AI API 키 얻기, 41
- 오픈AI API 키 활용 실습, 45
- API 사용량 및 요금 확인, 56

ChatGPT / 오픈AI

- 오픈AI와 ChatGPT 개요, 29
- GPT 모델 및 사용 사례, 33
- GPT-4o mini 대화 생성, 59

생성형 AI

- 생성형 AI 개요, 25
- 생성형 AI의 활용, 25
- 생성형 AI 모델의 아키텍처, 26
- 대표적인 생성형 AI 서비스, 26



구글 코랩

- 구글 코랩 사용 방법, 37

대화 생성 및 제어

- 대화 출력 길이 제어, 72
- AI 응답 제어, 76

프롬프트

- 프롬프트 엔지니어링 개요, 119
 - 프롬프트 체이닝, 127
 - 프롬프트 템플릿 최적화, 153
 - 프롬프트 파라미터 사용하기, 181

위스퍼

- 위스퍼 개요, 169
- 위스퍼 설치 및 기본 예제, 171
- 파이썬에서 위스퍼 SDK 사용하기, 173
- 위스퍼 텍스트 변환 최적화, 177–183

TTS (텍스트 음성 변환)

- 오픈 AI TTS 소개, 185
- 오픈 AI TTS 사용 방법, 186

CLIP

- CLIP 소개, 191
- CLIP 사용 방법, 191–199

DALL·E

- DALL·E 소개, 203
- 프롬프트 기반 이미지 생성 예제, 205–221
- DALL·E 이미지 편집, 236
- 이미지 변형 방법 및 사용 사례, 241–246

임베딩

- 임베딩의 의미, 251
- 임베딩 사용 사례, 252–254
- 텍스트 임베딩의 핵심 이해, 255
- 텍스트 임베딩 활용 예제, 259–277
- 고급 임베딩 예제, 279–316



파인 투닝

- 파인 투닝의 이해 및 구현, 323–348
- 파인 투닝 고급 예제 (정신 건강 코치), 353–366

문맥 및 기억력

- GPT의 기억력 및 문맥 문제, 369
- 대화 문맥 유지 및 해법, 370–378

부록

- 부록 A: 챗GPT 및 클로드 실습, 399–414
- 부록 B: 오픈AI 플레이그라운드 실습, 429–441
- 부록 C: 건강 상담 챗봇 구현, 445–452
- 부록 D: 작가 스타일 이미지 생성 앱, 460–466
- 부록 E: RAG 실습, 471–474
- 부록 F: 오토튠을 활용한 AI 에이전트, 485–486

이 예시는 주요 주제와 관련 하위 항목들을 중심으로 구성한 것입니다. 실제 인덱스 작성 시에는 독자
의 검색 편의성을 고려하여 추가 항목이나 세부 항목을 더 보완할 수 있습니다.

보통 인덱스는

CLIP소개 이런 식이 아니고 CLIP
RAG 실습이 아니고 RAG로 하지 않니?

몇 초 동안 이성 처리 완료 >

맞습니다. 인덱스는 일반적으로 항목명을 간결하게 작성하는 것이 좋습니다. 예를 들어, "CLIP 소개" 대
신 "CLIP"으로, "RAG 실습" 대신 "RAG"로 표시하는 식입니다. 독자가 핵심 단어를 빠르게 찾아볼 수 있
도록 단어 자체에 집중하는 것이 인덱스의 목적에 부합합니다.

간단한 예시를 아래에 수정해 보았습니다:

- **API:** 40, 41, 45, 56
- **ChatGPT:** 29, 33, 59
- **생성형 AI:** 25, 26
- **코랩:** 37
- **프롬프트:** 119, 127, 153, 181
- **위스퍼:** 169–173, 177–183
- **TTS:** 185, 186
- **CLIP:** 191–199
- **DALL-E:** 203–221, 236, 241–246
- **임베딩:** 251–254, 255–277, 279–316
- **파인 투닝:** 323–348, 353–366
- **문맥:** 369–378
- **RAG:** 471–474
- **오토젠:** 485–486
- **부록 A:** 399–414
- **부록 B:** 429–441
- **부록 C:** 445–452
- **부록 D:** 460–466



이와 같이 단일 용어로 표기함으로써 인덱스가 더 깔끔하고 사용하기 쉬워집니다. 추가로, 부연 설명이 필요한 경우 하위 항목으로 별도 처리하거나 페이지 번호로 보완하는 방식이 좋습니다.

좀 더 정확하는 페이지가 표시가 한페이지 단위로 되어야 해.

37초 동안 이성 처리 완료 >

아래는 목차에 나타난 각 항목의 페이지 번호를 단일 페이지 단위로 나열한 예시 인덱스입니다. 실제 페이지 번호는 편집 과정에서 최종 확인 후 조정해야 합니다.

API: 40, 41, 45, 46, 56

ChatGPT: 29, 33, 59

생성형 AI: 25, 26

코랩: 37

프롬프트: 119, 121, 127, 133, 138, 147, 153, 164

위스퍼: 169, 171, 173, 177, 179, 180, 181, 183

TTS: 185, 186

CLIP: 191, 192, 193, 199

DALL-E: 203, 205, 207, 208, 210, 213, 215, 219, 221, 223, 226, 227, 228, 236, 241, 246

임베딩: 251, 252, 253, 254, 255, 259, 260, 265, 277, 279, 296, 303, 308, 316

파인 투닝: 319, 320, 323, 331, 340, 344, 345, 346, 347, 348, 353, 355, 366

문맥: 369, 370, 373, 377, 378, 385

RAG: 471, 474

오토젠: 485, 486

부록 A: 399, 405, 411, 414

부록 B: 429, 434, 437, 441

부록 C: 445, 452

부록 D: 460, 462, 466

이 예시는 각 항목에 해당하는 페이지 번호를 한 페이지씩 나열한 것입니다. 인덱스 항목명은 가능한 한 간결하게, 핵심 단어(예: "CLIP", "RAG")만을 사용하여 독자가 원하는 내용을 빠르게 찾을 수 있도록 구성했습니다.

이 이후의 메시지는 사용자님에게만 나타납니다

ChatGPT에게 메시지를 쓰세요



ChatGPT는 실수를 할 수 있습니다. 중요한 정보를 확인하세요.

?