

서울R밋업 2023년

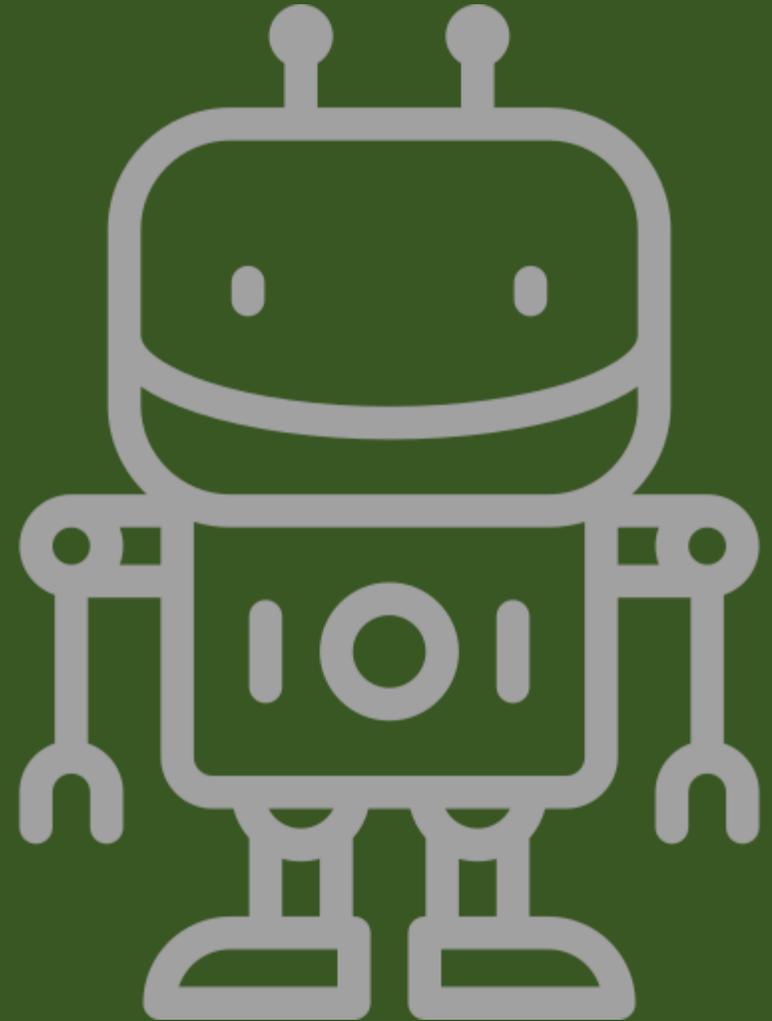
GPT와 임베딩을 이용한 분석 자동화

홍성학 euriion@gmail.com

WiderPlanet

Principal data scientist, ML/AI engineer

2023.8.10



순서

1. 분석자동화 시나리오
2. GPT 살펴보기
3. 임베딩
4. 벡터 검색
5. 분석자동화 만드는 요령

GPT와 임베딩을 활용한 자동분석

01

분석 자동화 시나리오

분석 자동화 시나리오

- 시나리오 유형1
 - 다양한 산업의 고객의 상황에 맞는 영업 데이터를 분석 (디지털 마케팅)
 - 예: 4000여개의 고객사의 데이터 분석을 통한 디지털 마케팅 가이드
- 시나리오 유형2
 - 회사 내의 플랫폼 또는 생산 시설의 데이터를 분석해서 플랫폼의 문제를 분석 (플랫폼 운영자)
 - 예: 분류 회사 C사의 유통 지연 문제, 상품 반품 문제
- 시나리오 유형3
 - 고객의 다양한 요구와 상황에 맞게 다양한 분석 결과를 서비스로 제공
 - 예: 자산 관리, 보험, 연금 등

분석 자동화 시나리오의 장벽

- GPT에 최신 자료 또는 특수한 일부 자료는 학습되어 있지 않음
 - 수 많은 자료를 매번 찾아서 프롬프트에 넣기는 너무 어려움
- 회사의 기밀 자료 또는 고객 자료를 외부에 유출 할 수 없음
 - 학습자료로 이용하도록 외부에 유출할 수 없음
- 자체 LLM을 구축하기는 너무 어려움



GPT와 임베딩을 활용한 자동분석

02

GPT 살펴보기

GPT : Generative Pre-trained Transformer

- GPT는 LLM
 - LLM = Large Language Model, 초거대 **언어** 모델
 - 언어 모델 , 계산 모델 **X**, 사고 모델 **X**, 추론 모델 **X**
 - 계산이 되는 듯 보이지만 그럴싸하게 결과를 생성할 뿐
- Generative : 생성에 특화된
- Pre-trained : 사전 학습된
- Transformer : 트랜스포머의 한 종류

구글 트랜스포머 모델의 생성 부분만 분리해서 만든
텍스트 생성 언어 모델

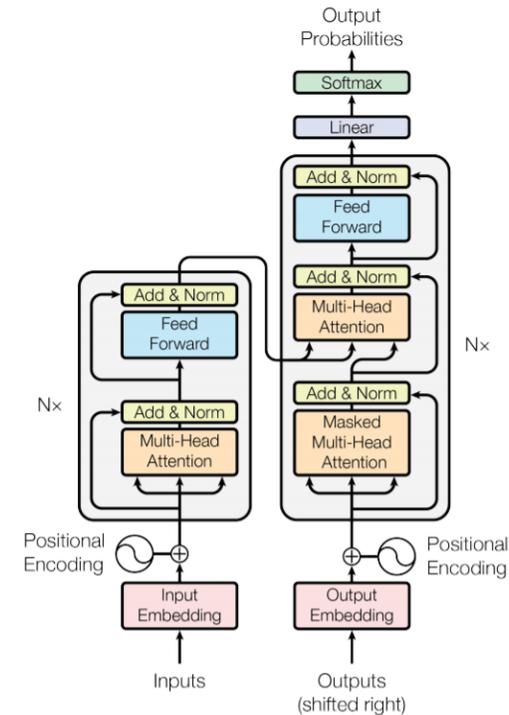


Figure 1: The Transformer - model architecture.

LLM의 생각의 사슬

- CoT, Chain of Thought
- 언어모델 (LLM)은 수리 능력 (산수), 추론 능력 (인과 추론)이 매우 떨어짐
- LLM 사실 생각하는 능력이 없음
- 질문을 단계별로 분할해서 답하도록 하면 추론을 하는 효과가 있음
- LLM은 확률론적 앵무새

CommonsenseQA	StrategyQA	Date Understanding	Sports Understanding
<p>Q: Sammy wanted to go to where the people were. Where might he go? Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock</p> <p>A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).</p>	<p>Q: Yes or no: Would a pear sink in water?</p> <p>A: The density of a pear is about 0.6 g/cm³, which is less than water. Thus, a pear would float. So the answer is no.</p>	<p>Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?</p> <p>A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.</p>	<p>Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."</p> <p>A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.</p>

OpenAI ChatGPT

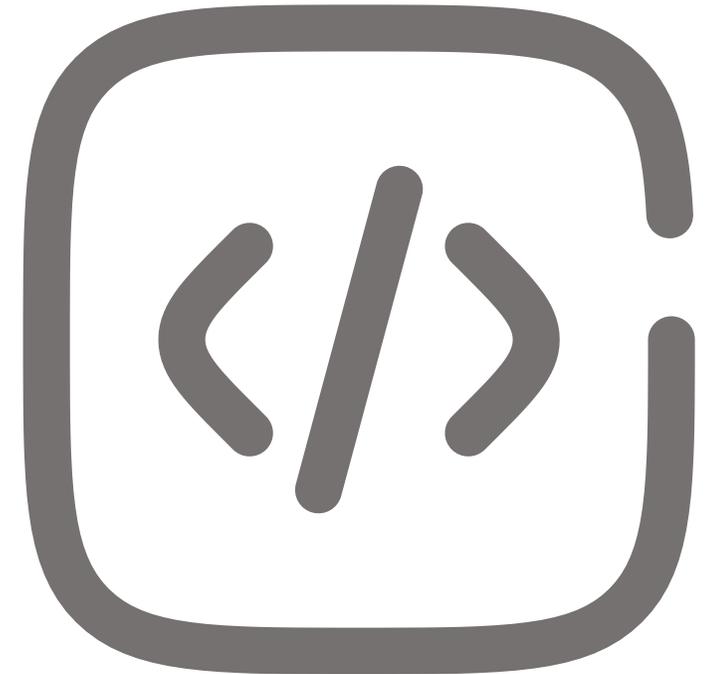
OpenAI사의 서비스 이름 또는 LLM 모델 이름

- ChatGPT model : GPT를 채팅용으로 파인 튜닝 한 것
- ChatGPT service : OpenAI사가 ChatGPT model을 이용해서 만든 채팅 서비스
- GPT ≠ ChatGPT, GPT와 ChatGPT는 다름



OpenAI CodeGPT

- CodeGPT : GPT로 소스 코드를 대량 학습 한 후 코드를 완성 할 수 있게 한 모델
- Github Copilot : CodeGPT로 만든 소스 코드 완성, 교정 서비스



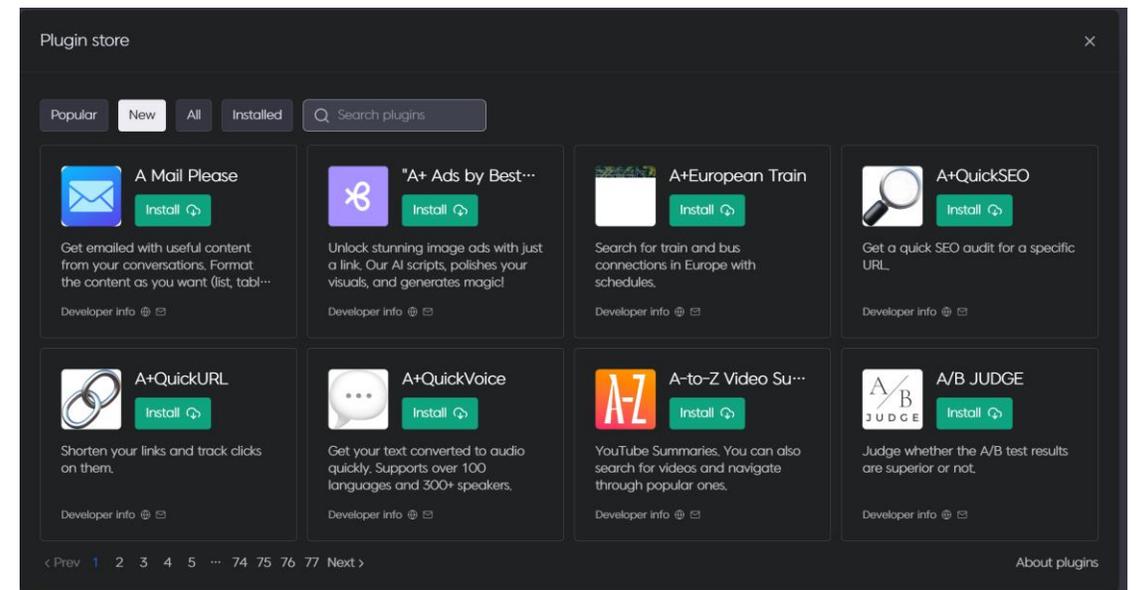
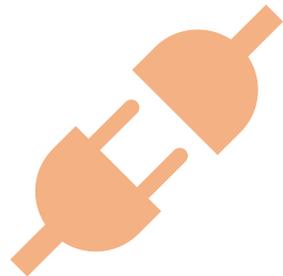
OpenAI Custom Instruction

- 미리 추가 해놓고 계속해서 사용하는 시스템 프롬프트
- 사용자의 반복적 프롬프트 입력을 간소화 한 것
- “당신은 데이터분석가입니다. 답은 한국어로만 합니다. 역사적 사실에 대해서 답을 하지 않습니다.”



OpenAI ChatGPT plugin

- ChatGPT의 외부 서비스와 연동한 확장 앱
- GPT function 기능을 활용했음
- 모델의 기능이 아닌 애플리케이션
- GPT function을 활용한 것



OpenAI ChatGPT code interpreter

- ChatGPT의 내부 확장 기능
- CodeGPT와 ChatGPT의 결합
- EDA를 비롯한 간단한 데이터 분석 수행
- 모델의 기능이 아닌 애플리케이션



GPT token

- 한글 및 한국의 형태소 분석과 유사
- Out of Vocabulary 문제, 매우 드물게 나오는 단어의 문제를 해결하기 위함
- 영어 1단어 = 약 2 ~ 3토큰
- 한글 1글자 = 약 2 ~ 3토큰
- UTF8 한글 1글자 = 3 바이트

GPT-3 Codex

Here is a sentence broken into tokens. Notice how most words are their own token, but that sometime names or complete words like ChatGPT, or misp~~ee~~led words, and , punctuation: , "are their" own tokens! As a rule of thumb for expressions in the English language there are approximately 750 words per 1000 tokens.

Clear

Show example

Tokens	Characters
68	312

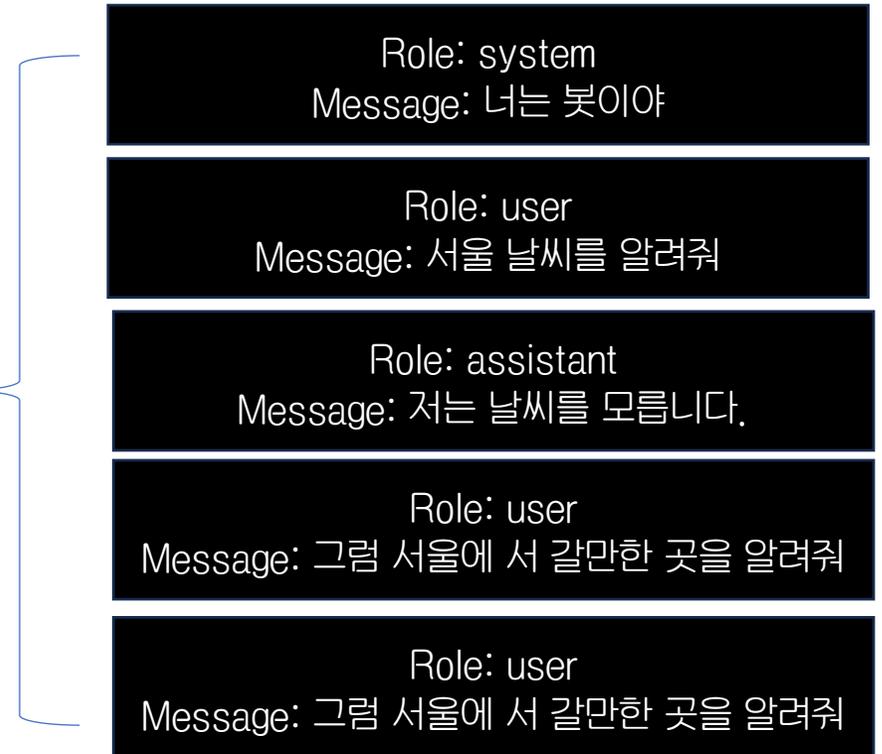
Here is a sentence broken into tokens. Notice how most words are their own token, but that sometime names or complete words like ChatGPT, or misp~~ee~~led words, and , punctuation: , "are their" own tokens! As a rule of thumb for expressions in the English language there are approximately 750 words per 1000 tokens.

TEXT TOKEN IDS

GPT context 컨텍스트

- 프롬프트와 답변을 합친 대화 세트
- 과거 대화 내역 + 입력 프롬프트 + 답변
- 토큰수 제한은 컨텍스트 단위로 제약이 있음
- 프롬프트가 매우 길면 매우 짧은 답변만 가능

컨텍스트



ChatGPT 모델 빌드 단계



Pre-trained
사전학습



Fine-tuning
학습



RLHF
강화학습



지식이 들어가는 단계

입력에 대한 완성을 만듦

모델 강화

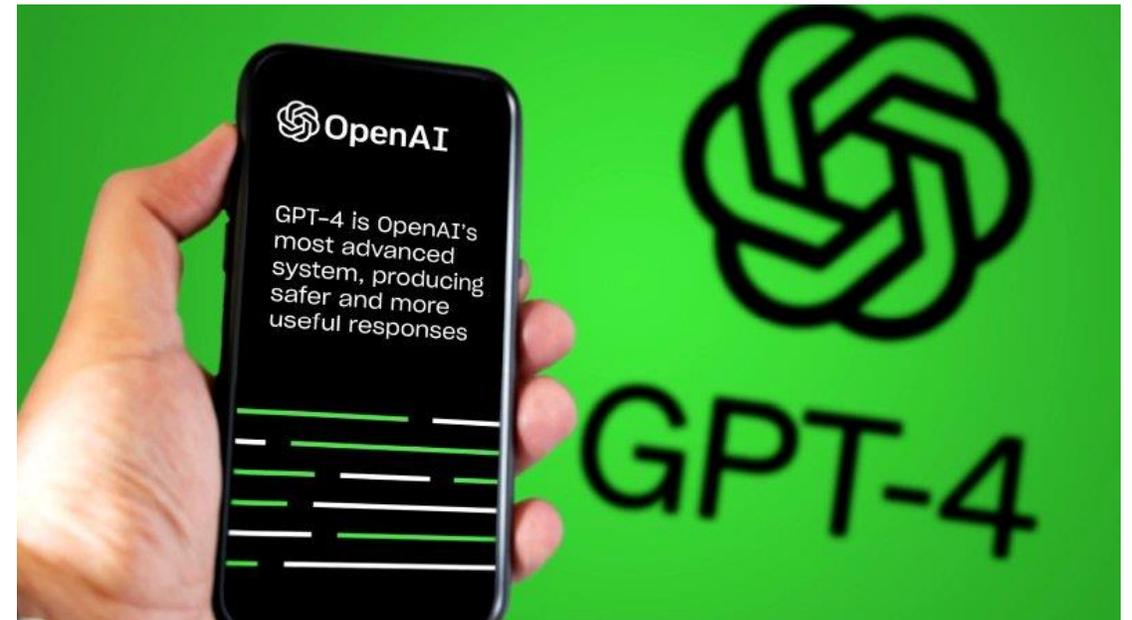
임베딩을 만들 수 있는
모델 생성

대답을 하는 모델을 생성

대답을 잘 하는 모델 생성

GPT 4

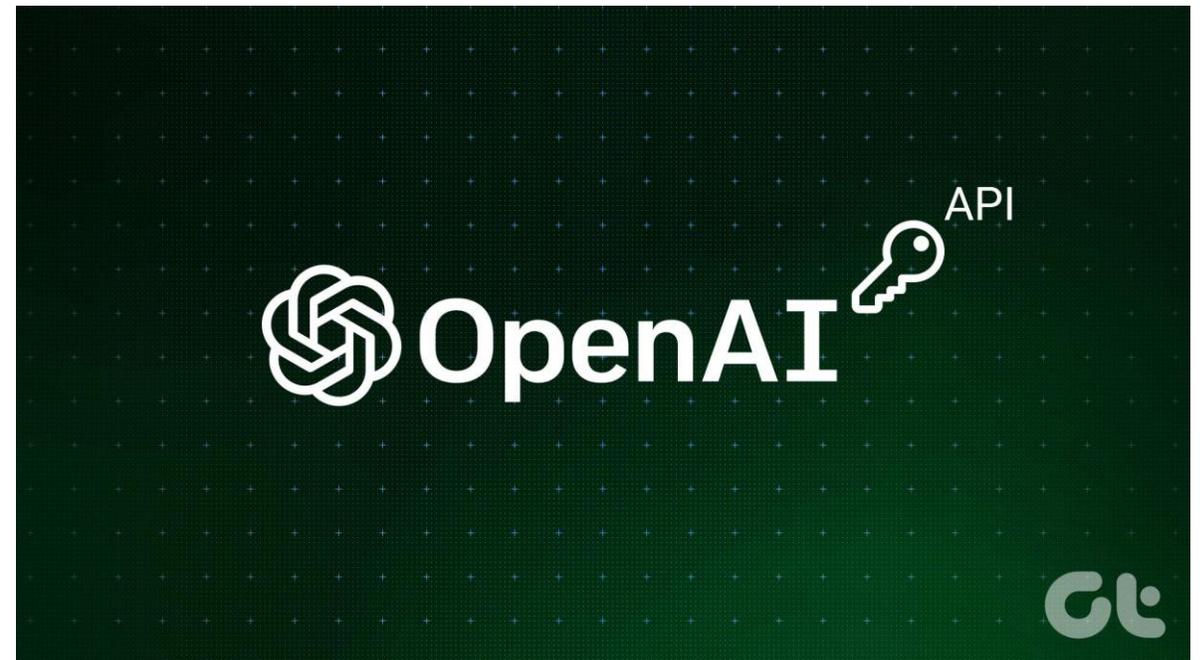
- << 파라미터 >>
 - 1조 7천억개
 - GPT3은 천7백억개
- << 모델 수 >>
 - 16개 (실제로는 2 ~ 3개 쓸 수도 있음)
 - 모델 1개당 1천억개 파라미터
 - 입력에 따라 모델 선택
- << GPU >>
 - H100S 25,000개로 100일 동안 학습
- << 사전학습 데이터 >>
 - 토큰 13조개 (책 1억권 분량)
 - 거의 대부분의 논문, Github 소스 전체, ...



OpenAI API

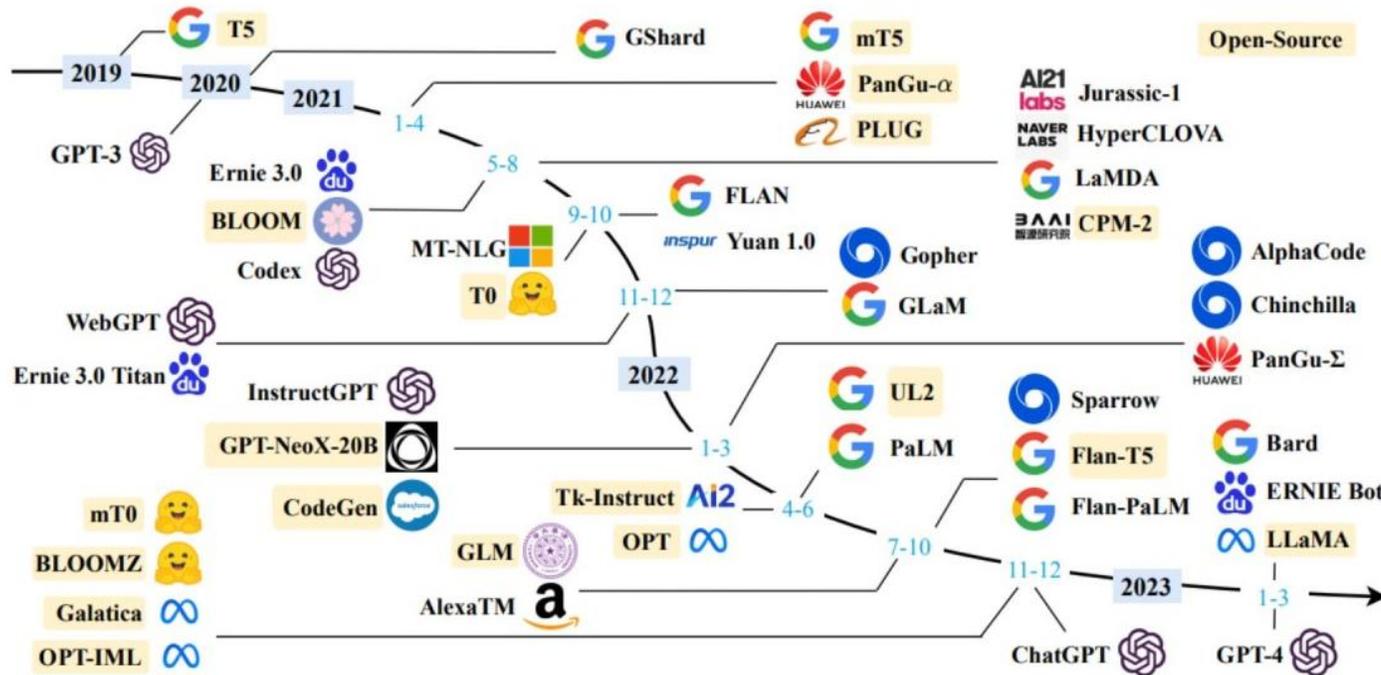
- GPT 모델을 온라인으로 사용하게 하는 API
- 지원하는 API가 ChatGPT와 동일하지 않으며 컨텍스트 관리를 별도 처리 필요

```
library(openai)
create_completion(
  model = "ada",
  prompt = "Generate a question
and an answer" )
#> $id
#> [1] "cml-
6MiImjcaCSuQYY6u8UA2Mm0rCdbEo"
#>
#> $object
#> [1] "text_completion"
```



GPT의 다른 대안 LLM?

- LLaMA2와 같은 최신 경량 LLM 모델은 GPT에 비해 낮음
- LLM은 모델 빌드하는데 시간, 인적 자원, 물적 자원 필요



그림출처: WTA

학습되지 않은 지식을 LLM에 넣는 방법?

- 전체 모델을 다시 생성
 - 기존 모델의 학습데이터 + 추가를 원하는 데이터
- 프롬프트에 데이터를 추가
 - 질문을 하기 전에 질문과 함께 지식을 넣어주고 답을 유도
- LoRA를 사용해서 부분학습
 - OpenAI는 아직 지원하지 않음
 - LLaMA 2는 지원

학습되지 않은 지식을 LLM에 넣기는 어려움

- LLM 모델을 처음부터 재생성해야 함
 - 많은 시간과 비용
- 프롬프트에 데이터를 추가하는 방법
 - 컨텍스트의 길이 제한
 - 많은 데이터를 넣을 수 없음
- LoRA를 사용해서 부분학습
 - OpenAI는 아직 지원하지 않음 (LLaMA 2는 지원)
 - 지식을 넣는 것과는 다른 작용

GPT와 임베딩을 활용한 자동분석

03

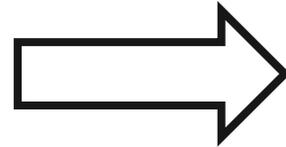
임베딩

임베딩 embedding

- 비정형 데이터를 낮은 차원의 벡터로 만드는 것

R은 통계 계산과 그래픽을 위한 프로그래밍 언어이자 소프트웨어 환경이다. 오픈소스로 쓰여졌으며 무료이다. 뉴질랜드 오클랜드 대학의 로버트 젠틀맨(Robert Gentleman)과 로스 이하카(Ross Ihaka)에 의해 시작되어 현재는 R 코어 팀이 개발하고 있다. R은 또한 GNU의 GPL 하에 배포되는 S 프로그래밍 언어의 구현으로 때때로 GNU S 로도 불린다. R은 통계 소프트웨어 개발과 자료 분석에 널리 사용되고 있으며, 패키지 개발이 용이해 통계 소프트웨어 개발에 많이 쓰이고 있다.

입력 단어



468954e-03, -2.997072e-02,
4.026989e-03, -3.114553e-02,
2.228877e-03, 1.320846e-03, -
1.290986e-02, -3.560328e-03, -
2.685747e-03,
(... 중간 생략...)
7.406221e-03, -3.624530e-03,
1.351586e-02, -1.047086e-02, -
2.099412e-02, 3.421530e-03, -
1.114535e-02, 2.742135e-03, -
2.090244e-02, -3.768594e-03

임베딩된 벡터

임베딩은?

- 비정형 데이터를 숫자로 바꿈
- 모델에 따라 크기가 다름
 - 예: OpenAI의 text-embedding-ada-002 모델은 1536 차원
- 빌드 모델에 따라 내용물이 완전히 다름
 - 동일한 구조로 모델을 재구성해도 내용물이 다르게 구성됨
- 모델 크기가 매우 큼
 - 벡터 그 자체를 문자열로 바꾸면 입력 데이터 보다 더 큰 사이즈를 차지함
 - 프롬프트에 벡터를 넣을 수 없음

OpenAI 임베딩 추출 코드

```
if (!require(remotes))
  install.packages("remotes")
require(remotes)

if (!require(openai))
  remotes::install_github("irudnyts/openai")
require(openai)

source("env.R", encoding = "UTF-8")
Sys.setenv(OPENAI_API_KEY = openai_token)

text ← c("R는 통계 계산과 그래픽을 위한 프로그래밍 언어이자 소프트웨어
환경이다. ",
        "오픈소스로 쓰여졌으며 무료이다. ",
        "뉴질랜드 오클랜드 대학의 로버트 젠틀맨(Robert Gentleman)과",
        "로스 이하카(Ross Ihaka)에 의해 시작되어 현재는 R 코어 팀이
개발하고 있다.",
        "R는 또한 GNU의 GPL 하에 배포되는 S 프로그래밍 언어의 구현으로
때때로 GNU S 로도 불린다.",
        "R는 통계 소프트웨어 개발과 자료 분석에 널리 사용되고 있으며,",
        "패키지 개발이 용이해 통계 소프트웨어 개발에 많이 쓰이고 있다.")

result ← create_embedding(
  model = "text-embedding-ada-002",
  input = text,
)

print(result$data$embedding[[1]])
```

임베딩의 문제

- 모델이 바뀔 때 마다 벡터가 바뀌므로 관리가 어려움
- 벡터 검색 엔진이 필요
 - $M \times M$ 의 유사도 측정을 매우 빨리 할 수 있는 엔진
- 임베딩 추출에도 빠른 처리를 위해서 GPU가 필요

GPT와 임베딩을 활용한 자동분석

04

벡터 검색

프롬프트에 데이터를 넣기 위한 준비물

데이터 플랫폼

- 자료 시스템, 문서 시스템(CMS), 데이터 베이스, 텍스트 검색 엔진, 벡터 검색 엔진, 이메일 시스템

데이터

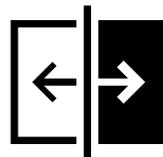
- 사내 용어집, 제품 홍보 문서, 사용자 매뉴얼, 고객응대 가이드라인, 시스템 로그, 소프트웨어 배포 로그, 채팅(대화) 로그

벡터 검색

- 자료를 검색하기 위한 2종류의 검색
 - 어휘 기반 검색, Lexical search, 텍스트 검색, Text search
 - 의미 기반 검색, Semantic search, 벡터 검색, Vector search

어휘 기반 검색

입력한 검색어에 매치되는 것을 찾음
빠르고 결과 명료함
가끔 의미와 거리가 먼 것을 찾음

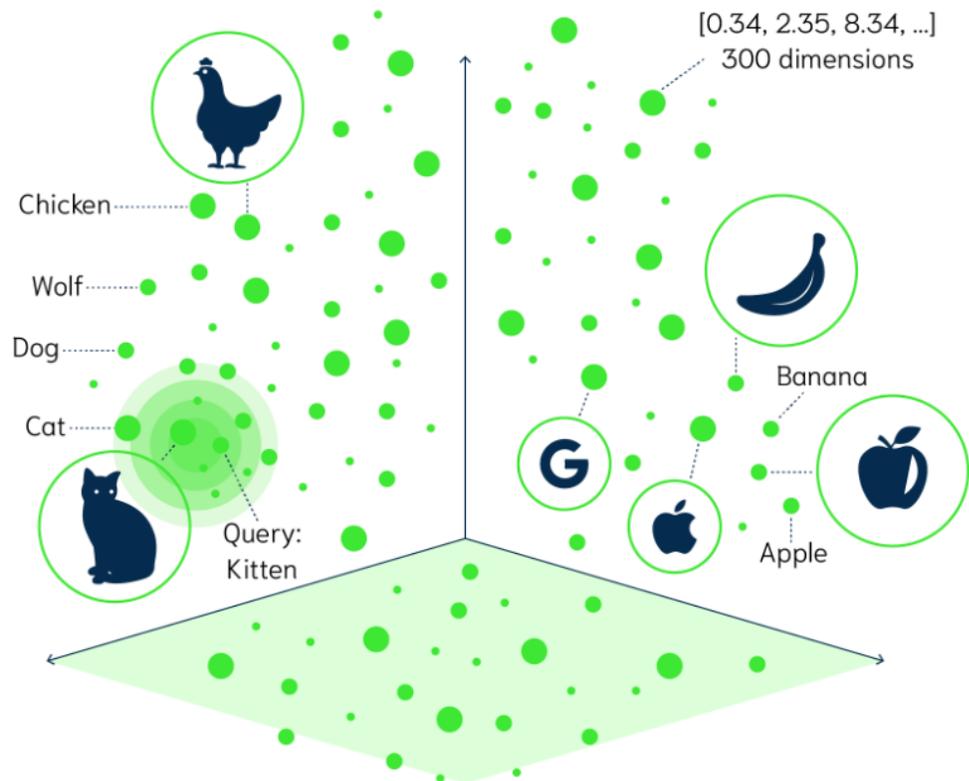


의미 기반 검색

입력한 벡터와 가장 가까운 것을 찾음
느리고 결과가 불명료
의미가 유사한 것을 찾음

벡터 검색이 필요한 이유

- 의미가 비슷한 내용을 찾는 것이 문장을 생성하는데 더 유용

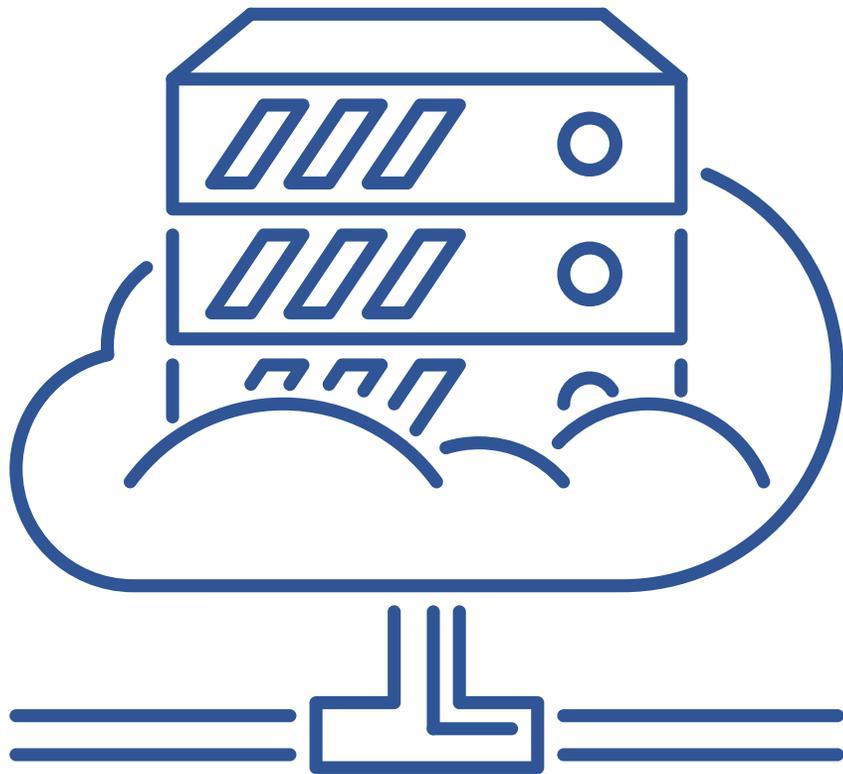


- 특정 책, 특정 보험 상품, 특정 금융 상품에 대한 질의를 했다면 명칭이 유사한 것보다는 내용이 비슷한 것을 사용하는 것이 더 합리적
 - ? 카레 만들기 = 카레의 기원?
 - ! 카레 만들기 = 인도 향신료
- ? LG 노트북 = LG 마우스
 - ! LG 노트북 = 삼성 테블릿

벡터 검색 엔진

- Pinecone : 온라인 서비스
- Qdrant : 자체 설치형
- Mivlus : 자체 설치
- Vespa : 자체 설치, 서비스
- Elasticsearch : 자체 설치, 서비스

서비스에 사용하기 위해서는
검색 속도가 매우 빠른 것이 필요



GPT와 임베딩을 활용한 자동분석

05

분석 자동화

GPT를 이용한 분석 자동화의 문제?

GPT와 데이터 분석의 문제점

- GPT는 논리적 추론을 하지 못함
 - ☞ 데이터를 해석하지 못함
- GPT의 컨텍스트 제한이 있음
 - ☞ 데이터를 모두 프롬프트에 넣지 못함

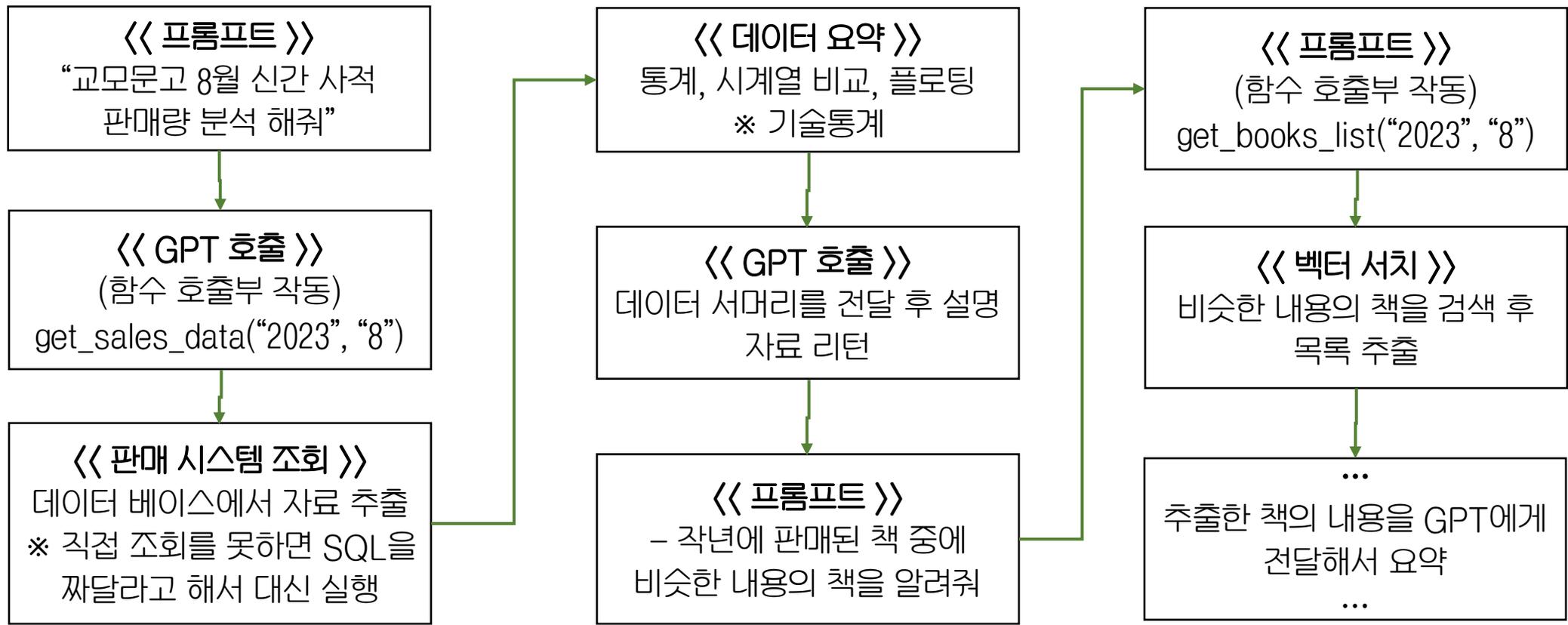
GPT는 데이터 분석을 못함



GPT를 이용한 분석 자동화 문제 풀기

- 데이터를 주지 않고 서머리를 주고 설명하게 함
- 데이터 요약, 통계적 테스트, 데이터 분석은 코드를 생성하게 하고 코드를 실행하게 함
- 플롯을 그리는 코드를 생성하게 하고 플롯을 생성함
- 필요한 데이터는 벡터 검색을 해서 프롬프트에 주입

GPT를 이용한 분석 자동화 흐름의 예



GPT function의 R 코드 예제

```

gpt_functions <- list(
  list("name" = "get_current_weather",
    "description" = "Get the current weather in a given location",
    "parameters" = list(
      "type" = "object",
      "properties" = list(
        "location" = list(
          "type" = "string",
          "description" = "The city and state, e.g. San Francisco, CA, 서울,
부산"
        ),
        "unit" = list(
          "type" = "string",
          "enum" = list("celsius", "fahrenheit")
        )
      ),
      "required" = list("location")
    )
  )
)

get_current_weather <- function(location, unit = "celsius") {
  url <- "https://api.openweathermap.org/data/2.5/weather"
  query <- list(
    "q" = location,
    "appid" = openweathermap_token,
    "units" = unit
  )
  response <- GET(url, query)
  content(response)
}

```

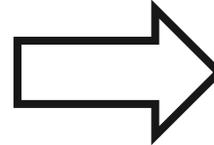
- GPT의 코드 학습의 능력(부산물)
- 함수의 정의를 제공하면 프롬프트에서 함수 호출부를 추출해서 리턴
- GPT가 함수를 실행하지는 않음 !
- 함수의 결과물을 GPT에게 전달하면 최종 결과물을 메시지로 리턴

계산기, 플러그인 모두 이 기능을 이용

GPT 분석 코드 생성의 예

```
prompt <- "generate a R language code to get statistics from a
CSV file for EDA
CSV file description
columns"
```

1. 상품번호, product id, integer
2. 분류, category, string
3. 상품명, product name, string
4. 가격, price, integer
5. 품절여부, sold out, string
6. 등록일, registered date, date



```
# Import necessary libraries
library(dplyr)
library(lubridate)

# Load the CSV file
data <- read.csv(file = \"Your_File_Path.csv\", header = TRUE,
fileEncoding = \"UTF-8-BOM\", skipNul = TRUE)

# Print a snippet of the data
head(data)

# Structure of the dataset
str(data)

# Convert 상품번호 and 가격 to numeric
data$상품번호 <- as.numeric(data$상품번호)
data$가격 <- as.numeric(data$가격)

# Convert 등록일 to date
data$등록일 <- dmy(data$등록일)

# Summary statistics of the numeric columns
summary(select_if(data, is.numeric))

# Frequency count of the categorical variables
table(data$분류)
table(data$상품명)
table(data$품절여부)

# Range of dates
range(data$등록일)
```

질문?

바보같은 질문은 없습니다.
부끄러워하지 말고 질문하세요.