OPENAI
Entity extraction for long documents

김도형

Entity extraction for long documents
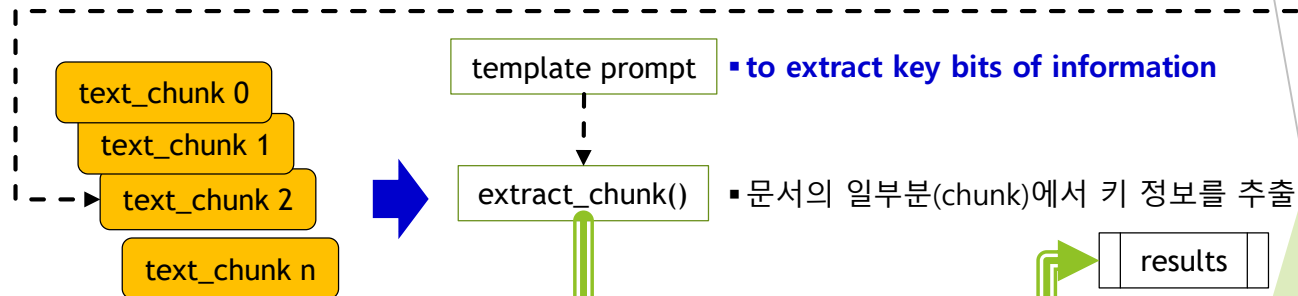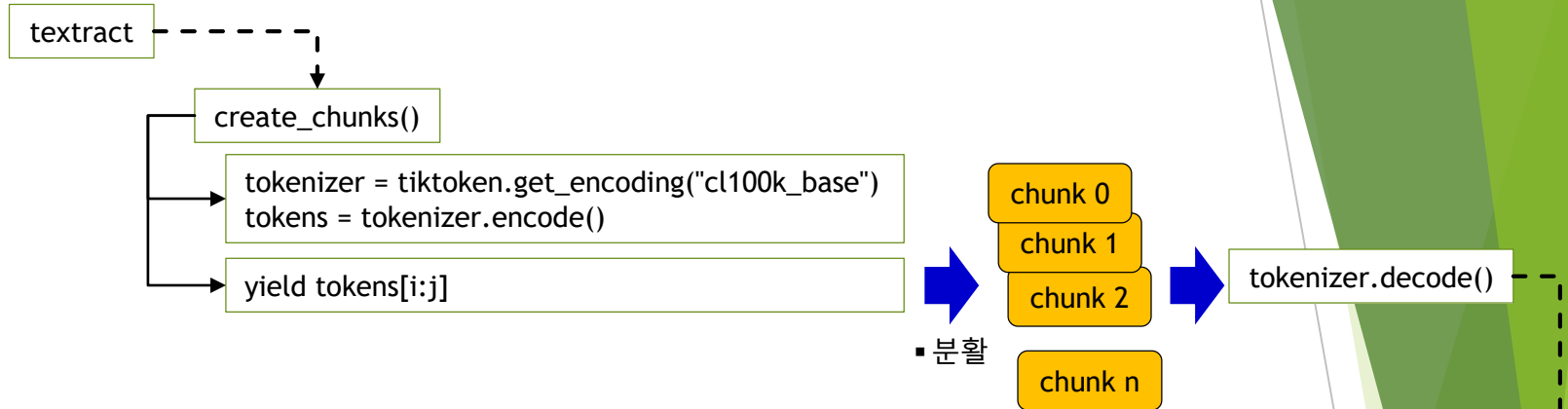
PDF

textract

create_chunks()

```
tokenizer = tiktoken.get_encoding("cl100k_base")
tokens = tokenizer.encode()
```

yield tokens[i:j]

chunk 0
chunk 1
chunk 2
chunk n

tokenizer.decode()

▪ 분활

text_chunk 0
text_chunk 1
text_chunk 2
text_chunk n

template prompt

▪ **to extract key bits of information**

extract_chunk()

▪ 문서의 일부분(chunk)에서 키 정보를 추출
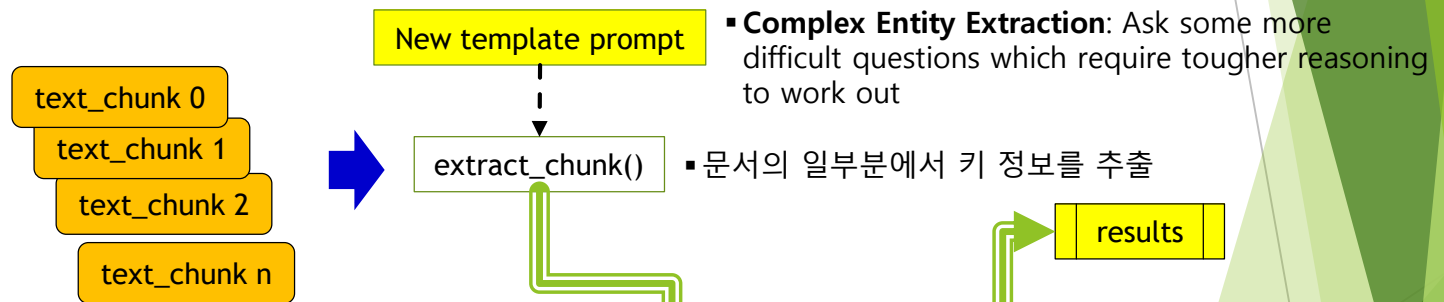
results

OpenAI

```
response = openai.Completion.create(
    model='text-davinci-003',
    prompt=prompt,
    temperature=0,
    max_tokens=1500,
    top_p=1,
    frequency_penalty=0,
    presence_penalty=0
    )
```

▪ 문서의 일부분(document = chunk)과 프롬프트 템플릿을 입력으로 받아, 프롬프트와 결합하여 OpenAI의 GPT-3 모델을 사용하여 응답을 생성

Entity extraction for long documents

New template prompt

text_chunk 0
text_chunk 1
text_chunk 2
text_chunk n

extract_chunk()

- **Complex Entity Extraction**: Ask some more difficult questions which require tougher reasoning to work out

- 문서의 일부분에서 키 정보를 추출

results

OpenAI

```
response = openai.Completion.create(
    model='text-davinci-003',
    prompt=prompt,
    temperature=0,
    max_tokens=1500,
    top_p=1,
    frequency_penalty=0,
    presence_penalty=0
    )
```

LLM based Chatbot
to query
Private Knowledge Base

Private Knowledge Base

Documentation

Chat Interface

Ask a question....

Submit

1.

2.

Embedding Model

5.

Query

- Text Chunk
- Stored Vectors
- Embedded Query Context
- Retrieved Vectors

2.

5.

9.

Vector
[0.01, ... , 0.43]

Vector
[0.41, ... , 0.02]

Answer

3.

6.

Indexing

Vector Index

Prompt

Answer the Query.
Only use Context
to construct the answer.

7.

9.

Approximate Nearest Neighbour search (ANN)

9.

9.

Vector Database

- Embedding(Latent) Space

LLM