Generating Answers: Input Text ⇒ Chunking ⇒ Embedding ⇒ Search Index ⇒ Query ⇒ Search ⇒ Question ⇒ Answer

Khoi Duong
Prof. Chang
CS589
4/10/2024

2. Generating Answers: Input Text ==> Chunking ==> Embedding ==> Search Index ==> Query ==> Search ==> Question ==> Answer .
    - Project Implementation
        - Generating Answers
            - References
    - Process for the project documentation
        - Step 1: Adding the project to your portofolio
            1. Please use Google Slides to document the project
                - Copy from a Google Slides file and mofigy the file, but still keep the original Google Slides file.
            2. Please link your presentation on GitHub using this structure

```
Generative AI
  - Fine-Tuning
    + Generating Answers
```

        - Step 2: Submit
            1. The URLs of the Google Slides and GitHub web pages related to this project.
            2. A PDF file of your Google Slides

Project implementation

- Step 1: Preprocessing input texts
- Step 1.1: Split into a list of paragraphs
- Step 1.2: Clean up to remove empty spaces and new line
- Step 2: Embeddings
- Step 2.1: Chunking: Get the embeddings (vectors) from input texts
- Step 2.2: Build a search index from embeddings (vectors)
- Step 2.2.1: Check the dimensions of the embeddings
- Step 2.2.2: Create the search index, pass the size of embeddings(size of vector)
- Step 2.3: Add all the vectors to the search index
- Step 3: Searching Articles
- Step 4: Generating Answers
- Step 4.1: Generating Answers - Test Case 1
- Step 4.2: Generating Answers - Test Case 2
- Step 4.3: Generating Answers - Test Case 3

Get env variable needed for ReRank

Before we start, we need to set up the environment and get the env variable for the program, including WEAVIATE_API_KEY, WEAVIATE_API_URL, and COHERE_API_KEY

We start by installing cohere and weaviate-client with pip:

- pip install cohere
- pip install weaviate-client
- pip install annoy

We can also use pip to install any missing modules later on when running the program.

```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589$ pip install cohere
Collecting cohere
  Downloading cohere-5.2.5-py3-none-any.whl (150 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 150.6/150.6 KB 1.2 MB/s eta 0:00:00
Requirement already satisfied: requests<3.0.0,>=2.0.0 in ./venv/lib/python3.10/site-packages (from
 cohere) (2.31.0)
Requirement already satisfied: types-requests<3.0.0,>=2.0.0 in ./venv/lib/python3.10/site-packages
 (from cohere) (2.31.0.20240218)
Collecting fastavro<2.0.0,>=1.9.4
  Downloading fastavro-1.9.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.1/3.1 MB 3.6 MB/s eta 0:00:00
Requirement already satisfied: typing_extensions>=4.0.0 in ./venv/lib/python3.10/site-packages (fr
om cohere) (4.10.0)
Requirement already satisfied: pydantic>=1.9.2 in ./venv/lib/python3.10/site-packages (from cohere
) (1.10.14)
Requirement already satisfied: tokenizers<0.16.0,>=0.15.2 in ./venv/lib/python3.10/site-packages (
from cohere) (0.15.2)
```

```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589$ pip install weaviate-client
Collecting weaviate-client
  Downloading weaviate_client-4.5.5-py3-none-any.whl (306 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 306.8/306.8 KB 1.1 MB/s eta 0:00:00
Collecting grpcio-tools<2.0.0,>=1.57.0
  Downloading grpcio_tools-1.62.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.8
MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.8/2.8 MB 117.4 kB/s eta 0:00:00
Collecting validators==0.22.0
  Downloading validators-0.22.0-py3-none-any.whl (26 kB)
Collecting grpcio-health-checking<2.0.0,>=1.57.0
  Downloading grpcio_health_checking-1.62.1-py3-none-any.whl (18 kB)
Requirement already satisfied: requests<3.0.0,>=2.30.0 in ./venv/lib/python3.10/site-packages (fro
m weaviate-client) (2.31.0)
Requirement already satisfied: httpx==0.27.0 in ./venv/lib/python3.10/site-packages (from weaviate
-client) (0.27.0)
Requirement already satisfied: grpcio<2.0.0,>=1.57.0 in ./venv/lib/python3.10/site-packages (from
weaviate-client) (1.62.0)
Collecting authlib<2.0.0,>=1.2.1
  Downloading Authlib-1.3.0-py2.py3-none-any.whl (223 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 223.7/223.7 KB 79.9 kB/s eta 0:00:00
Collecting pydantic<3.0.0,>=2.5.0
```

```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ pip install annoy
Collecting annoy
  Downloading annoy-1.17.3.tar.gz (647 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 647.5/647.5 KB 5.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Using legacy 'setup.py install' for annoy, since package 'wheel' is not insta
lled.
Installing collected packages: annoy
  Running setup.py install for annoy ... done
Successfully installed annoy-1.17.3
```

Getting API key from Cohere and Weaviate

In order to get the API key from Cohere and Weaviate, refer to the documentation [Week 10 HW 3 - CS589 - Khoi Duong - 19610](#) to go through the process.

```
COHERE_API_KEY=NMqLBcUcNlFX1BPJaej8F0P2hyTemU
WEAVIATE_API_KEY=77oVYq7lBPNuZaT5iluYrmyPH5Tm
WEAVIATE_API_URL=https://rerank-ui45fbli.weav
```

Source code

FineTuning > generating_ans.py > ...

```python
question = "Are side projects important when you \
            are starting to learn about AI?"

text = """
The rapid rise of AI has led to a rapid rise \
in AI jobs, and \
many people are building exciting careers \
in this field. A \
career is a decades-long journey, and the \
path is not always \
straightforward. Over many years, I've \
been privileged to see \
thousands of students as well as engineers \
in companies large \
and small navigate careers in AI. In this \
and the next few letters, \
I'd like to share a few thoughts that \
might be useful in \
charting your own course.

Three key steps of career growth are \
learning (to gain technical \
and other skills), working on projects \
(to deepen skills, build \
a portfolio, and create impact) and \
searching for a job. \
These steps stack on top of each other:

Initially, you focus on gaining \
foundational technical skills.
After having gained foundational skills, \
you lean into project \
work. During this period, you'll \
```

FineTuning > generating_ans.py > ...

```python
course of your career
, so you'll have ample opportunity to \
refine your
thinking on what's worthwhile. Given \
the huge number
of possible AI projects, rather than the \
conventional "ready
, aim, fire" approach, you can accelerate your
progress with "ready, fire, aim."
"""

import os
# read local .env file
from dotenv import load_dotenv, find_dotenv
_ = load_dotenv(find_dotenv())

import cohere
import numpy as np
import warnings
warnings.filterwarnings('ignore')

# Step 1. Preprocessing input texts
# Step 1.1 Split into a list of paragraphs
texts = text.split('\n\n')
# Step 1.2 Clean up to remove empty spaces and
#          new lines
texts = np.array([t.strip(' \n') for t in
                  texts if t])

texts[:3]
```

Source code (cont)

```python
# Step 2. Embeddings
co = cohere.Client(os.environ['COHERE_API_KEY'])
# Step 2.1 Chunking: Get the embeddings (vectors)
#                    from input texts
response = co.embed(
    texts=texts.tolist(),
).embeddings
# Step 2.2 Use AnnoyIndex( to build a search index
#          from the embeddings (vectors)
from annoy import AnnoyIndex
import numpy as np
import pandas as pd
# Step 2.2.1 Check the dimensions of the
#            embeddings
embeds = np.array(response)
# Step 2.2.2 Create the search index, pass the
#            size of embedding (vector)
search_index = AnnoyIndex(embeds.shape[1],
                          'angular')
# Step 2.3 Add all the vectors to the search index
for i in range(len(embeds)):
    search_index.add_item(i, embeds[i])

# 10 trees
search_index.build(10)
search_index.save('test.ann')
```

```python
# Step 3. Searching Articles
def search_andrews_article(query):
    # Get the query's embedding
    query_embed = co.embed(texts=[query]).embeddings

    # Retrieve the nearest neighbors
    similar_item_ids = search_index.get_nns_by_vector(
                query_embed[0],
                10,
                include_distances=True)

    search_results = texts[similar_item_ids[0]]

    return search_results

results = search_andrews_article(
    "Are side projects a good idea when trying \
     to build a career in AI?"
)

print(results[0])
```

Source code (cont)

```python
# Step 4. Generating Answers
def ask_andrews_article(question, num_generations=1):

    # Search the text archive
    results = search_andrews_article(question)

    # Get the top result
    context = results[0]

    # Prepare the prompt
    prompt = f"""
    Excerpt from the article titled "How to
    Build a Career in AI"
    by Andrew Ng:
    {context}
    Question: {question}

    Extract the answer of the question from
    the text provided.
    If the text doesn't contain the answer,
    reply that the answer is not available."""

    prediction = co.generate(
        prompt=prompt,
        max_tokens=70,
        model="command-nightly",
        temperature=0.5,
        num_generations=num_generations
    )

    return prediction.generations
```

```python
# Step 4.1 Generating Answers - Test Case 1
results = ask_andrews_article("Are side projects a good idea when \
                              trying to build a career in AI?",)

print(results[0])

# Step 4.2 Generating Answers - Test Case 2
results = ask_andrews_article("Are side projects a good idea when \
                              trying to build a career in AI?", num_generations=3)

for gen in results:
    print(gen)
    print('--')

# Step 4.3 Generating Answers - Test Case 3
results = ask_andrews_article(
    "What is the most viewed televised event?",
    num_generations=5
)

for gen in results:
    print(gen)
    print('--')
```

Run the program

Here is the output of the program:

```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ python generating_ans.py
Join existing projects. If you find someone else with an
idea, ask to join their project.
Keep reading and talking to people. I come up with
new ideas whenever I spend a lot of time reading,
taking courses, or talking with domain experts. I’
m confident that you will, too.
Focus on an application area. Many researchers are trying to
advance basic AI technology – say, by inventing the next
generation of transformers or further scaling up
language models – so, while this is an exciting
direction, it is hard.
But the variety of applications to which machine learning has
not yet been applied is vast! I’m fortunate
to have been able to apply neural networks to everything from
autonomous helicopter flight to online advertising, partly
because I jumped in when relatively few people were working on
those applications. If your company or school cares about a
particular application, explore the possibilities for machine
learning. That can give you a first look at a potentially
creative application – one where you can do unique work –
that no one else has done yet.
Develop a side hustle. Even if you have a full-time job, a
fun project that may or may not develop into something bigger
can stir the creative juices and strengthen bonds with collaborators.
When I was a full-time professor, working on
online education wasn’t part of my “job” (which was doing
research and teaching classes). It was a fun hobby that I
often worked on out of passion for education. My early
experiences recording videos at home helped me later in
working on online education in a more substantive way.
Silicon Valley abounds with stories of startups that started as side
projects. So long as it doesn’t create a conflict with your
employer, these projects can be a stepping stone to something
significant.  Given a few project ideas, which one should
you jump into?
```

```
Here’s a quick checklist of factors to consider:
id='a190b43b-c27e-45fc-bef0-f94109181114' text='Yes.' index=None likelihood=None token_likelihoods=None finish_reason='COMPLETE'
id='f16a8668-fef9-4973-8dbe-2d11f7f9e320' text='Yes, side projects are a good idea when trying to build a career in AI, as they can stir creative juices, strengthe
n bonds with collaborators, and potentially lead to something significant.' index=None likelihood=None token_likelihoods=None finish_reason='COMPLETE'
--
id='a0aa291b-efda-4f78-b6f1-b60d7132f467' text='Yes, side projects are a good idea when trying to build a career in AI, as they can stir creative juices, strengthe
n bonds with collaborators, and potentially lead to something significant.' index=None likelihood=None token_likelihoods=None finish_reason='COMPLETE'
--
id='527572f6-a1c1-4a6f-a2c1-f83187126bd9' text='Yes, side projects are a good idea when trying to build a career in AI, as they can stir creative juices and streng
then bonds with collaborators.' index=None likelihood=None token_likelihoods=None finish_reason='COMPLETE'
--
id='c94986c9-cfc3-4229-8aa2-7bbf7de6d84c' text='The answer is not available.' index=None likelihood=None token_likelihoods=None finish_reason='COMPLETE'
--
id='8bebef73-c534-4f45-92a1-23bc9cd7a0f1' text='The answer is not available.' index=None likelihood=None token_likelihoods=None finish_reason='COMPLETE'
--
id='a29863e8-e057-4520-a2b1-c380fb561b66' text='The answer is not available.' index=None likelihood=None token_likelihoods=None finish_reason='COMPLETE'
--
id='9ec67a0b-2290-4440-a0b9-ef42050671b8' text='The answer is not available.' index=None likelihood=None token_likelihoods=None finish_reason='COMPLETE'
--
id='083f30e5-098c-47cc-99a6-3a5a4d48b168' text='The answer is not available.' index=None likelihood=None token_likelihoods=None finish_reason='COMPLETE'
--
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$
```

Reference

- Problem
- Generating Answers

Original repo: https://github.com/MynameisKoi/CS589/tree/main/FineTuning

Source code:

- generating_ans.py