Thomas Cowart

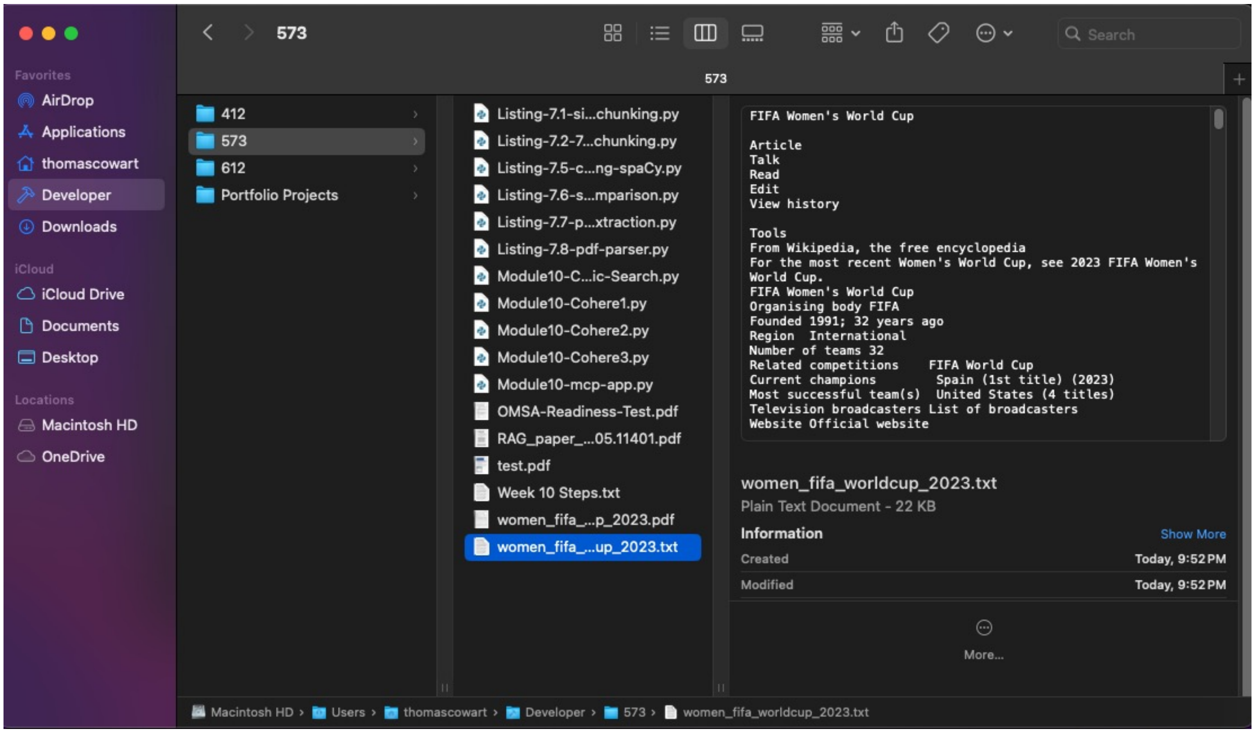
Prof. Gill

ISYS 573

4/7/25

Week 10 HW

1.


573
Search
573
Favorites
AirDrop
Applications
thomascowart
Developer
Downloads
iCloud
iCloud Drive
Documents
Desktop
Locations
Macintosh HD
OneDrive
412
573
612
Portfolio Projects
Listing-7.1-si...chunking.py
Listing-7.2-7...chunking.py
Listing-7.5-c...ng-spaCy.py
Listing-7.6-s...mparison.py
Listing-7.7-p...xtraction.py
Listing-7.8-pdf-parser.py
Module10-C...ic-Search.py
Module10-Cohere1.py
Module10-Cohere2.py
Module10-Cohere3.py
Module10-mcp-app.py
OMSA-Readiness-Test.pdf
RAG_paper_...05.11401.pdf
test.pdf
Week 10 Steps.txt
women_fifa_...p_2023.pdf
women_fifa_...up_2023.txt
More...
Macintosh HD › Users › thomascowart › Developer › 573 › women_fifa_worldcup_2023.txt

```
FIFA Women's World Cup

Article
Talk
Read
Edit
View history

Tools
From Wikipedia, the free encyclopedia
For the most recent Women's World Cup, see 2023 FIFA Women's
World Cup.
FIFA Women's World Cup
Organising body FIFA
Founded 1991; 32 years ago
Region  International
Number of teams 32
Related competitions    FIFA World Cup
Current champions        Spain (1st title) (2023)
Most successful team(s)  United States (4 titles)
Television broadcasters List of broadcasters
Website Official website
```

women_fifa_worldcup_2023.txt

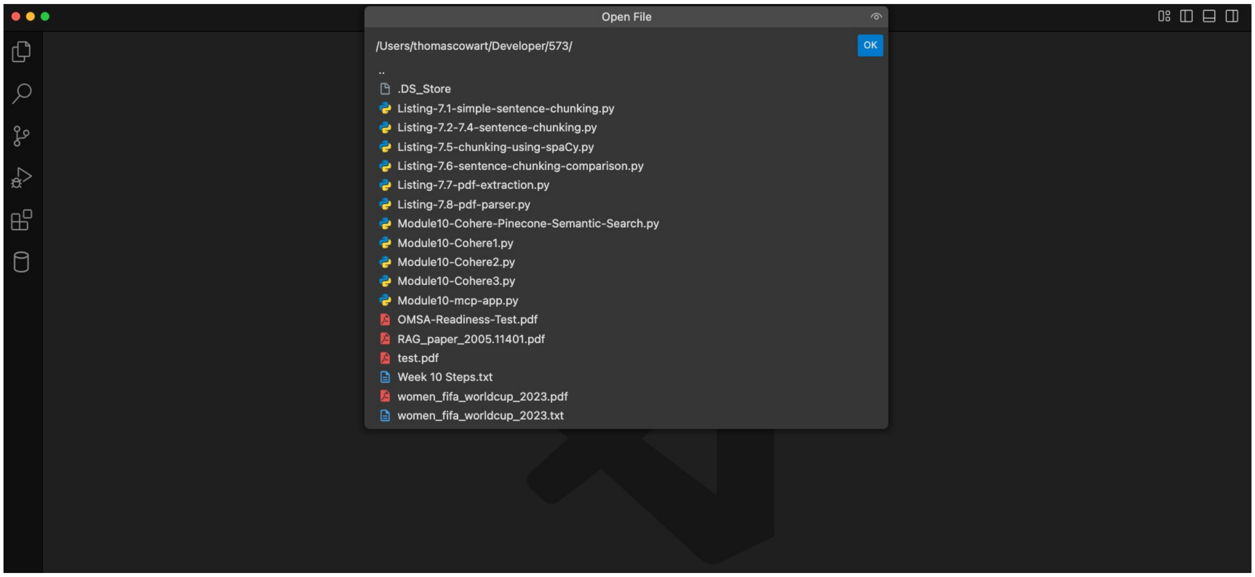
Plain Text Document - 22 KB

Information — Show More

Created — Today, 9:52 PM

Modified — Today, 9:52 PM

2.

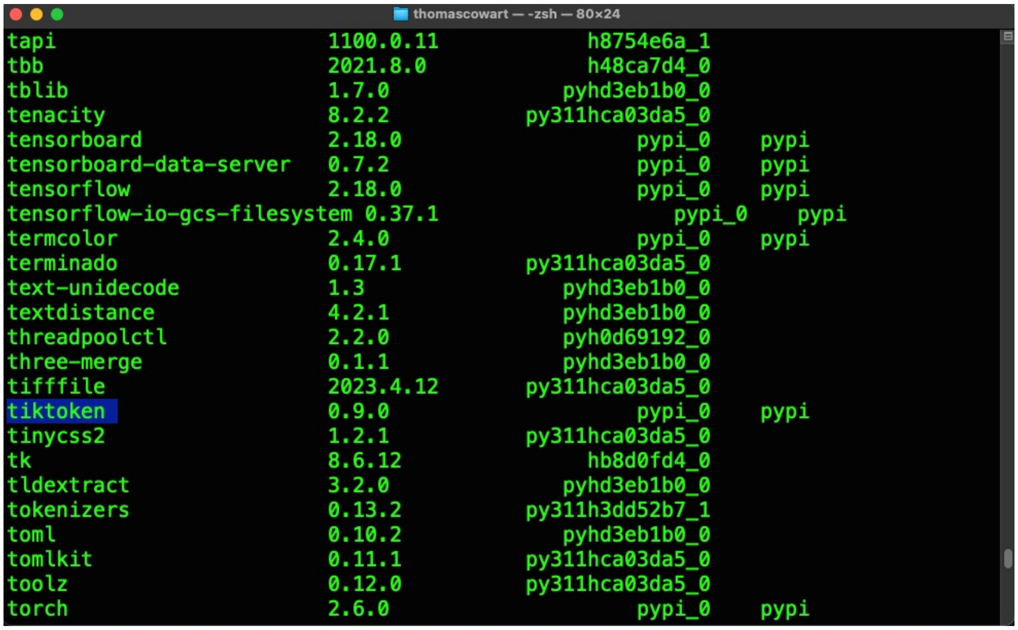


/Users/thomascowart/Developer/573/

..
.DS_Store
Listing-7.1-simple-sentence-chunking.py
Listing-7.2-7.4-sentence-chunking.py
Listing-7.5-chunking-using-spaCy.py
Listing-7.6-sentence-chunking-comparison.py
Listing-7.7-pdf-extraction.py
Listing-7.8-pdf-parser.py
Module10-Cohere-Pinecone-Semantic-Search.py
Module10-Cohere1.py
Module10-Cohere2.py
Module10-Cohere3.py
Module10-mcp-app.py
OMSA-Readiness-Test.pdf
RAG_paper_2005.11401.pdf
test.pdf
Week 10 Steps.txt
women_fifa_worldcup_2023.pdf
women_fifa_worldcup_2023.txt

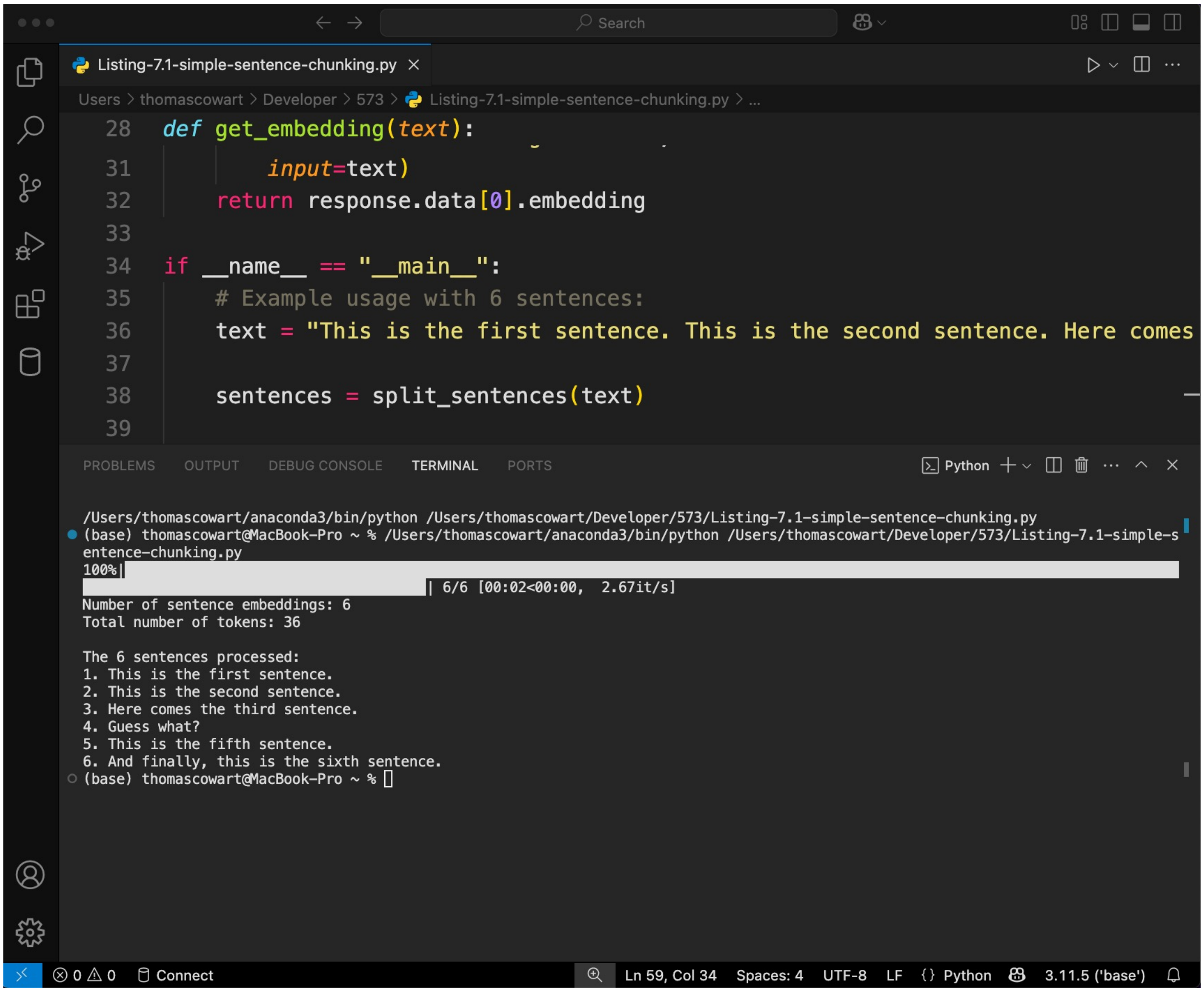
3.

4.



```
tapi                      1100.0.11            h8754e6a_1
tbb                       2021.8.0             h48ca7d4_0
tblib                     1.7.0              pyhd3eb1b0_0
tenacity                  8.2.2           py311hca03da5_0
tensorboard               2.18.0                   pypi_0    pypi
tensorboard-data-server   0.7.2                    pypi_0    pypi
tensorflow                2.18.0                   pypi_0    pypi
tensorflow-io-gcs-filesystem 0.37.1                   pypi_0    pypi
termcolor                 2.4.0                    pypi_0    pypi
terminado                 0.17.1          py311hca03da5_0
text-unidecode            1.3                pyhd3eb1b0_0
textdistance              4.2.1              pyhd3eb1b0_0
threadpoolctl             2.2.0              pyh0d69192_0
three-merge               0.1.1              pyhd3eb1b0_0
tifffile                  2023.4.12       py311hca03da5_0
tiktoken                  0.9.0                    pypi_0    pypi
tinycss2                  1.2.1           py311hca03da5_0
tk                        8.6.12               hb8d0fd4_0
tldextract                3.2.0              pyhd3eb1b0_0
tokenizers                0.13.2          py311h3dd52b7_1
toml                      0.10.2             pyhd3eb1b0_0
tomlkit                   0.11.1          py311hca03da5_0
toolz                     0.12.0          py311hca03da5_0
torch                     2.6.0                    pypi_0    pypi
```

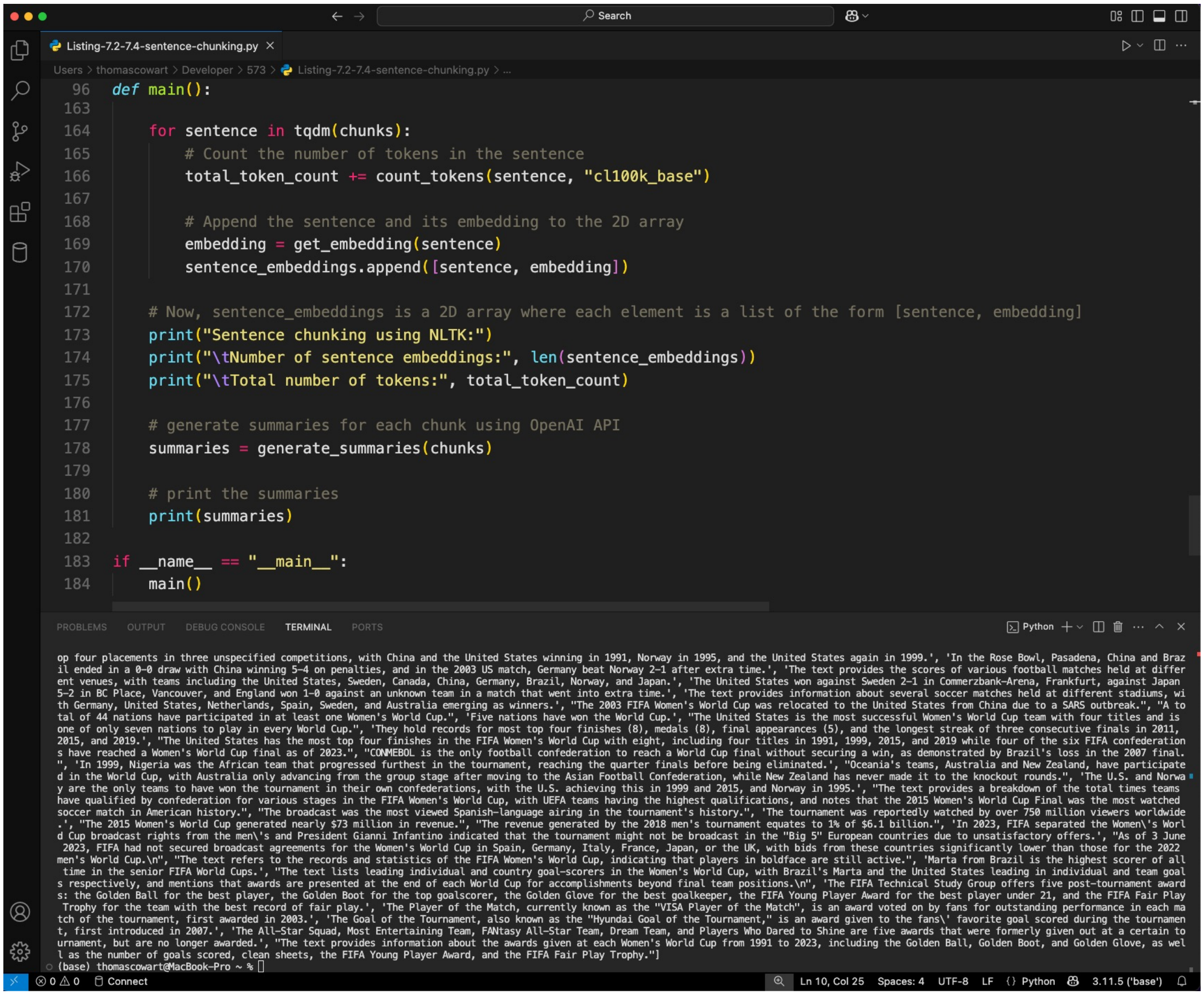
4.


Search
Listing-7.1-simple-sentence-chunking.py
Users > thomascowart > Developer > 573 > Listing-7.1-simple-sentence-chunking.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
0 0
Connect
Ln 59, Col 34
Spaces: 4
UTF-8
LF
{ } Python
3.11.5 ('base')

```python
def get_embedding(text):
        input=text)
    return response.data[0].embedding

if __name__ == "__main__":
    # Example usage with 6 sentences:
    text = "This is the first sentence. This is the second sentence. Here comes

    sentences = split_sentences(text)

```

```
/Users/thomascowart/anaconda3/bin/python /Users/thomascowart/Developer/573/Listing-7.1-simple-sentence-chunking.py
(base) thomascowart@MacBook-Pro ~ % /Users/thomascowart/anaconda3/bin/python /Users/thomascowart/Developer/573/Listing-7.1-simple-s
entence-chunking.py
100%|
                                         | 6/6 [00:02<00:00,  2.67it/s]
Number of sentence embeddings: 6
Total number of tokens: 36

The 6 sentences processed:
1. This is the first sentence.
2. This is the second sentence.
3. Here comes the third sentence.
4. Guess what?
5. This is the fifth sentence.
6. And finally, this is the sixth sentence.
(base) thomascowart@MacBook-Pro ~ %
```

5.



Search
Listing-7.2-7.4-sentence-chunking.py
Users > thomascowart > Developer > 573 > Listing-7.2-7.4-sentence-chunking.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
0 0
Connect
Ln 10, Col 25
Spaces: 4
UTF-8
LF
{} Python
3.11.5 ('base')

```python
def main():

    for sentence in tqdm(chunks):
        # Count the number of tokens in the sentence
        total_token_count += count_tokens(sentence, "cl100k_base")

        # Append the sentence and its embedding to the 2D array
        embedding = get_embedding(sentence)
        sentence_embeddings.append([sentence, embedding])

    # Now, sentence_embeddings is a 2D array where each element is a list of the form [sentence, embedding]
    print("Sentence chunking using NLTK:")
    print("\tNumber of sentence embeddings:", len(sentence_embeddings))
    print("\tTotal number of tokens:", total_token_count)

    # generate summaries for each chunk using OpenAI API
    summaries = generate_summaries(chunks)

    # print the summaries
    print(summaries)

if __name__ == "__main__":
    main()
```

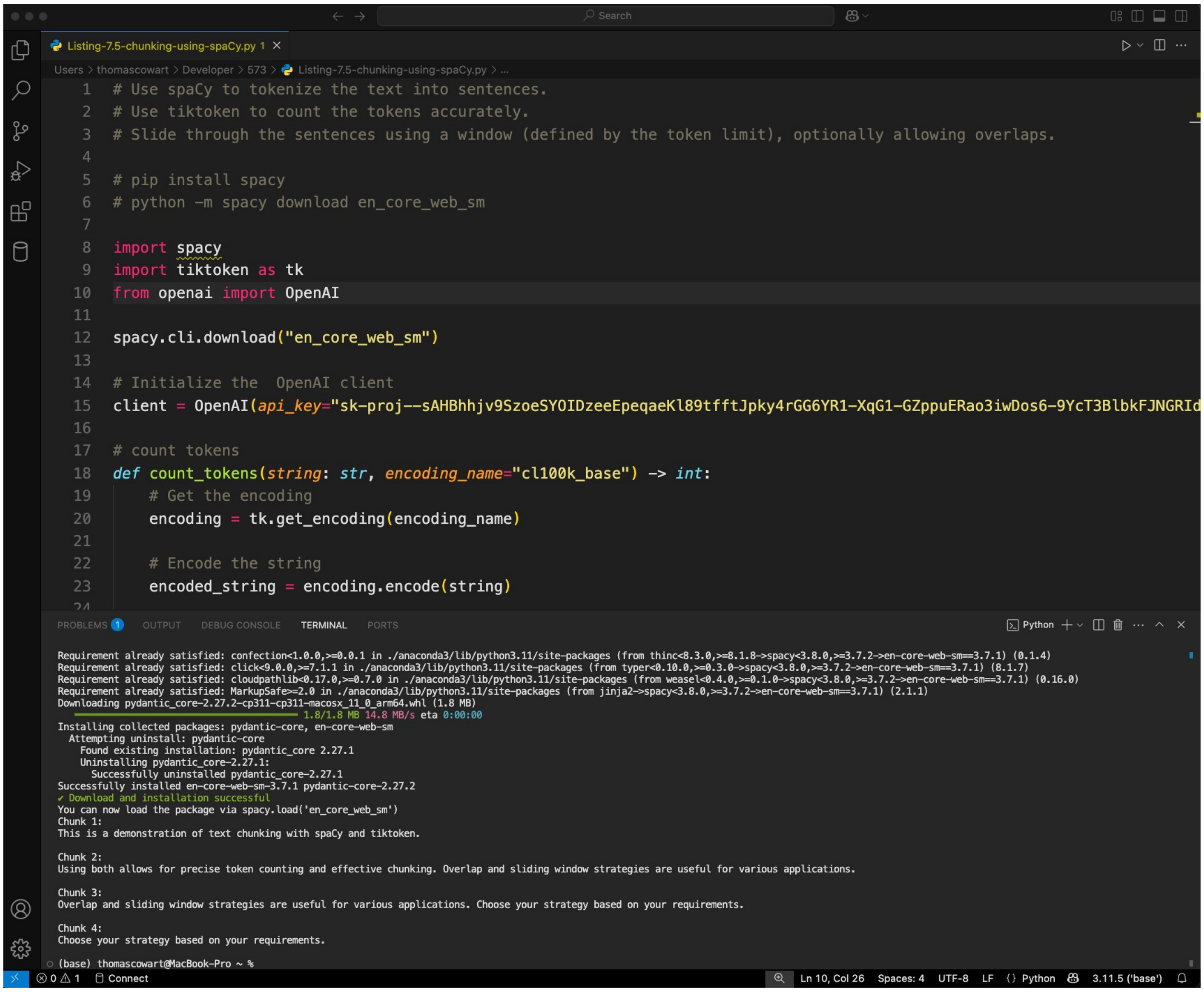
```
op four placements in three unspecified competitions, with China and the United States winning in 1991, Norway in 1995, and the United States again in 1999.', 'In the Rose Bowl, Pasadena, China and Braz
il ended in a 0-0 draw with China winning 5-4 on penalties, and in the 2003 US match, Germany beat Norway 2-1 after extra time.', 'The text provides the scores of various football matches held at differ
ent venues, with teams including the United States, Sweden, Canada, China, Germany, Brazil, Norway, and Japan.', 'The United States won against Sweden 2-1 in Commerzbank-Arena, Frankfurt, against Japan
5-2 in BC Place, Vancouver, and England won 1-0 against an unknown team in a match that went into extra time.', 'The text provides information about several soccer matches held at different stadiums, wi
th Germany, United States, Netherlands, Spain, Sweden, and Australia emerging as winners.', "The 2003 FIFA Women's World Cup was relocated to the United States from China due to a SARS outbreak.", "A to
tal of 44 nations have participated in at least one Women's World Cup.", 'Five nations have won the World Cup.', "The United States is the most successful Women's World Cup team with four titles and is
one of only seven nations to play in every World Cup.", 'They hold records for most top four finishes (8), medals (8), final appearances (5), and the longest streak of three consecutive finals in 2011,
2015, and 2019.', "The United States has the most top four finishes in the FIFA Women's World Cup with eight, including four titles in 1991, 1999, 2015, and 2019 while four of the six FIFA confederation
s have reached a Women's World Cup final as of 2023.", "CONMEBOL is the only football confederation to reach a World Cup final without securing a win, as demonstrated by Brazil's loss in the 2007 final.
", 'In 1999, Nigeria was the African team that progressed furthest in the tournament, reaching the quarter finals before being eliminated.', "Oceania's teams, Australia and New Zealand, have participate
d in the World Cup, with Australia only advancing from the group stage after moving to the Asian Football Confederation, while New Zealand has never made it to the knockout rounds.", 'The U.S. and Norwa
y are the only teams to have won the tournament in their own confederations, with the U.S. achieving this in 1999 and 2015, and Norway in 1995.', "The text provides a breakdown of the total times teams
have qualified by confederation for various stages in the FIFA Women's World Cup, with UEFA teams having the highest qualifications, and notes that the 2015 Women's World Cup Final was the most watched
soccer match in American history.", "The broadcast was the most viewed Spanish-language airing in the tournament's history.", 'The tournament was reportedly watched by over 750 million viewers worldwide
.', "The 2015 Women's World Cup generated nearly $73 million in revenue.", "The revenue generated by the 2018 men's tournament equates to 1% of $6.1 billion.", 'In 2023, FIFA separated the Women\'s Worl
d Cup broadcast rights from the men\'s and President Gianni Infantino indicated that the tournament might not be broadcast in the "Big 5" European countries due to unsatisfactory offers.', "As of 3 June
 2023, FIFA had not secured broadcast agreements for the Women's World Cup in Spain, Germany, Italy, France, Japan, or the UK, with bids from these countries significantly lower than those for the 2022
men's World Cup.\n", "The text refers to the records and statistics of the FIFA Women's World Cup, indicating that players in boldface are still active.", 'Marta from Brazil is the highest scorer of all
 time in the senior FIFA World Cups.', "The text lists leading individual and country goal-scorers in the Women's World Cup, with Brazil's Marta and the United States leading in individual and team goal
s respectively, and mentions that awards are presented at the end of each World Cup for accomplishments beyond final team positions.\n", 'The FIFA Technical Study Group offers five post-tournament award
s: the Golden Ball for the best player, the Golden Boot for the top goalscorer, the Golden Glove for the best goalkeeper, the FIFA Young Player Award for the best player under 21, and the FIFA Fair Play
 Trophy for the team with the best record of fair play.', 'The Player of the Match, currently known as the "VISA Player of the Match", is an award voted on by fans for outstanding performance in each ma
tch of the tournament, first awarded in 2003.', 'The Goal of the Tournament, also known as the "Hyundai Goal of the Tournament," is an award given to the fans\' favorite goal scored during the tournamen
t, first introduced in 2007.', 'The All-Star Squad, Most Entertaining Team, FANtasy All-Star Team, Dream Team, and Players Who Dared to Shine are five awards that were formerly given out at a certain to
urnament, but are no longer awarded.', "The text provides information about the awards given at each Women's World Cup from 1991 to 2023, including the Golden Ball, Golden Boot, and Golden Glove, as wel
l as the number of goals scored, clean sheets, the FIFA Young Player Award, and the FIFA Fair Play Trophy."]
(base) thomascowart@MacBook-Pro ~ %
```

Simple: Manually splits text by spaces or punctuation. It’s basic and doesn’t account for structure or meaning.

Textwrap: Splits text into lines of a specific width. It’s useful for formatting text to fit within a set number of characters.

NLTK: More advanced, using natural language processing to split text based on sentence or word boundaries. It’s smarter about structure and context.

6.


Search
Listing-7.5-chunking-using-spaCy.py 1
Users > thomascowart > Developer > 573 > Listing-7.5-chunking-using-spaCy.py > ...
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
0 1
Connect
Ln 10, Col 26
Spaces: 4
UTF-8
LF
{} Python
3.11.5 ('base')

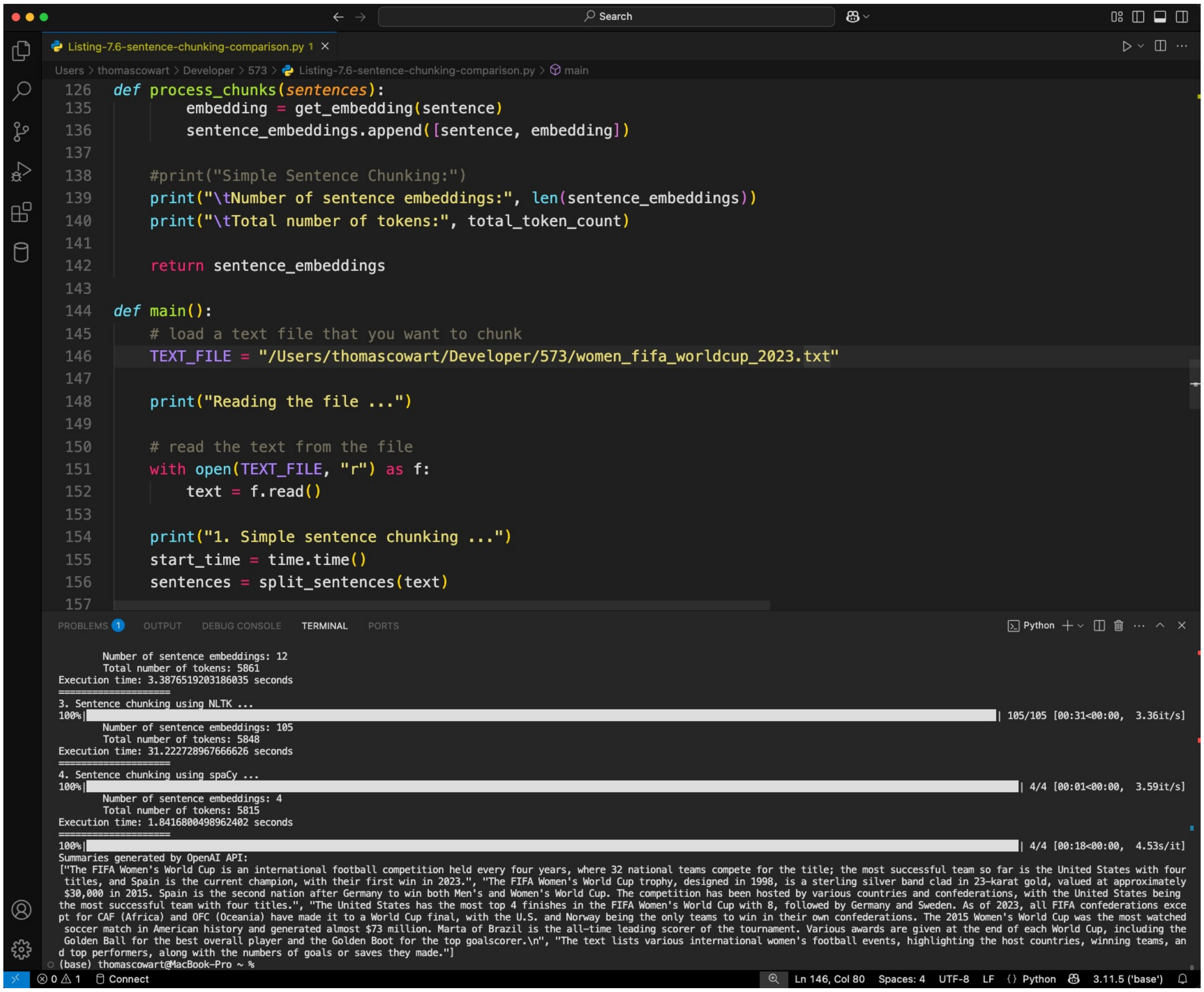
```python
# Use spaCy to tokenize the text into sentences.
# Use tiktoken to count the tokens accurately.
# Slide through the sentences using a window (defined by the token limit), optionally allowing overlaps.

# pip install spacy
# python -m spacy download en_core_web_sm

import spacy
import tiktoken as tk
from openai import OpenAI

spacy.cli.download("en_core_web_sm")

# Initialize the  OpenAI client
client = OpenAI(api_key="sk-proj--sAHBhhjv9SzoeSYOIDzeeEpeqaeKl89tfftJpky4rGG6YR1-XqG1-GZppuERao3iwDos6-9YcT3BlbkFJNGRId

# count tokens
def count_tokens(string: str, encoding_name="cl100k_base") -> int:
    # Get the encoding
    encoding = tk.get_encoding(encoding_name)

    # Encode the string
    encoded_string = encoding.encode(string)
```

```
Requirement already satisfied: confection<1.0.0,>=0.0.1 in ./anaconda3/lib/python3.11/site-packages (from thinc<8.3.0,>=8.1.8->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.1.4)
Requirement already satisfied: click<9.0.0,>=7.1.1 in ./anaconda3/lib/python3.11/site-packages (from typer<0.10.0,>=0.3.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (8.1.7)
Requirement already satisfied: cloudpathlib<0.17.0,>=0.7.0 in ./anaconda3/lib/python3.11/site-packages (from weasel<0.4.0,>=0.1.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in ./anaconda3/lib/python3.11/site-packages (from jinja2->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.1.1)
Downloading pydantic_core-2.27.2-cp311-cp311-macosx_11_0_arm64.whl (1.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 14.8 MB/s eta 0:00:00
Installing collected packages: pydantic-core, en-core-web-sm
  Attempting uninstall: pydantic-core
    Found existing installation: pydantic_core 2.27.1
    Uninstalling pydantic_core-2.27.1:
      Successfully uninstalled pydantic_core-2.27.1
Successfully installed en-core-web-sm-3.7.1 pydantic-core-2.27.2
✔ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
Chunk 1:
This is a demonstration of text chunking with spaCy and tiktoken.

Chunk 2:
Using both allows for precise token counting and effective chunking. Overlap and sliding window strategies are useful for various applications.

Chunk 3:
Overlap and sliding window strategies are useful for various applications. Choose your strategy based on your requirements.

Chunk 4:
Choose your strategy based on your requirements.

(base) thomascowart@MacBook-Pro ~ %
```

spaCy is faster and built for real world, large-scale applications with pre-trained models for tasks like NER and “part of speech” tagging. It’s easy to use and performance focused.

NLTK is more research-oriented, offering lots of flexibility and educational tools, but it’s slower and requires more setup. It’s great for learning and experimenting with NLP.

7.


Search
Listing-7.6-sentence-chunking-comparison.py 1
Users > thomascowart > Developer > 573 > Listing-7.6-sentence-chunking-comparison.py > main
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
0 ⚠ 1
Connect
Ln 146, Col 80
Spaces: 4
UTF-8
LF
{} Python
3.11.5 ('base')

```python
def process_chunks(sentences):
        embedding = get_embedding(sentence)
        sentence_embeddings.append([sentence, embedding])

    #print("Simple Sentence Chunking:")
    print("\tNumber of sentence embeddings:", len(sentence_embeddings))
    print("\tTotal number of tokens:", total_token_count)

    return sentence_embeddings

def main():
    # load a text file that you want to chunk
    TEXT_FILE = "/Users/thomascowart/Developer/573/women_fifa_worldcup_2023.txt"

    print("Reading the file ...")

    # read the text from the file
    with open(TEXT_FILE, "r") as f:
        text = f.read()

    print("1. Simple sentence chunking ...")
    start_time = time.time()
    sentences = split_sentences(text)

```

```
        Number of sentence embeddings: 12
        Total number of tokens: 5861
Execution time: 3.3876519203186035 seconds
====================
3. Sentence chunking using NLTK ...
100%|████████████████████████████████████████████████████████████| 105/105 [00:31<00:00,  3.36it/s]
        Number of sentence embeddings: 105
        Total number of tokens: 5848
Execution time: 31.222728967666626 seconds
====================
4. Sentence chunking using spaCy ...
100%|██████████████████████████████████████████████████████████████| 4/4 [00:01<00:00,  3.59it/s]
        Number of sentence embeddings: 4
        Total number of tokens: 5815
Execution time: 1.8416800498962402 seconds
====================
100%|██████████████████████████████████████████████████████████████| 4/4 [00:18<00:00,  4.53s/it]
Summaries generated by OpenAI API:
["The FIFA Women's World Cup is an international football competition held every four years, where 32 national teams compete for the title; the most successful team so far is the United States with four
 titles, and Spain is the current champion, with their first win in 2023.", "The FIFA Women's World Cup trophy, designed in 1998, is a sterling silver band clad in 23-karat gold, valued at approximately
 $30,000 in 2015. Spain is the second nation after Germany to win both Men's and Women's World Cup. The competition has been hosted by various countries and confederations, with the United States being
the most successful team with four titles.", "The United States has the most top 4 finishes in the FIFA Women's World Cup with 8, followed by Germany and Sweden. As of 2023, all FIFA confederations exce
pt for CAF (Africa) and OFC (Oceania) have made it to a World Cup final, with the U.S. and Norway being the only teams to win in their own confederations. The 2015 Women's World Cup was the most watched
 soccer match in American history and generated almost $73 million. Marta of Brazil is the all-time leading scorer of the tournament. Various awards are given at the end of each World Cup, including the
 Golden Ball for the best overall player and the Golden Boot for the top goalscorer.\n", "The text lists various international women's football events, highlighting the host countries, winning teams, an
d top performers, along with the numbers of goals or saves they made."]
(base) thomascowart@MacBook-Pro ~ %
```

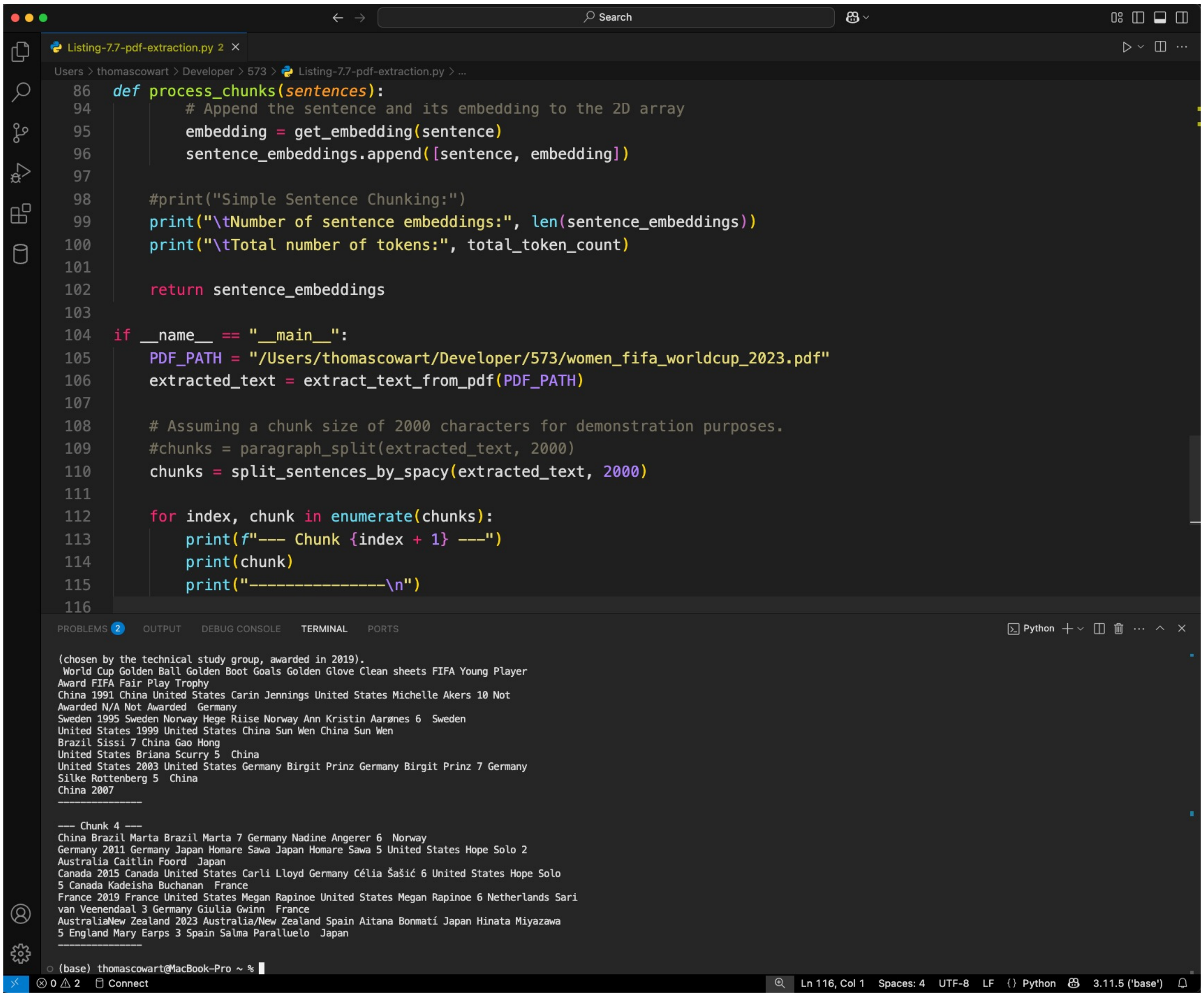
Here’s how I’d rank the chunking methods by speed:

1. Simple – Fastest. It just splits text based on spaces or basic delimiters.
2. Textwrap – Slightly slower than simple because it considers line length and formatting, but still pretty fast.
3. spaCy – Fast, especially for production-level tasks. It’s optimized but has more overhead than simple methods.
4. NLTK – Slowest. NLTK provides flexibility but requires more processing due to its complexity and range of tools.

Choice:

I'd choose simple for tasks that don't require deep NLP understanding. It's the fastest and works well for basic tasks like splitting text by spaces or punctuation. If I need more nuanced chunking (like sentences or words), spaCy is a solid choice for its speed and pre-trained models.

8.



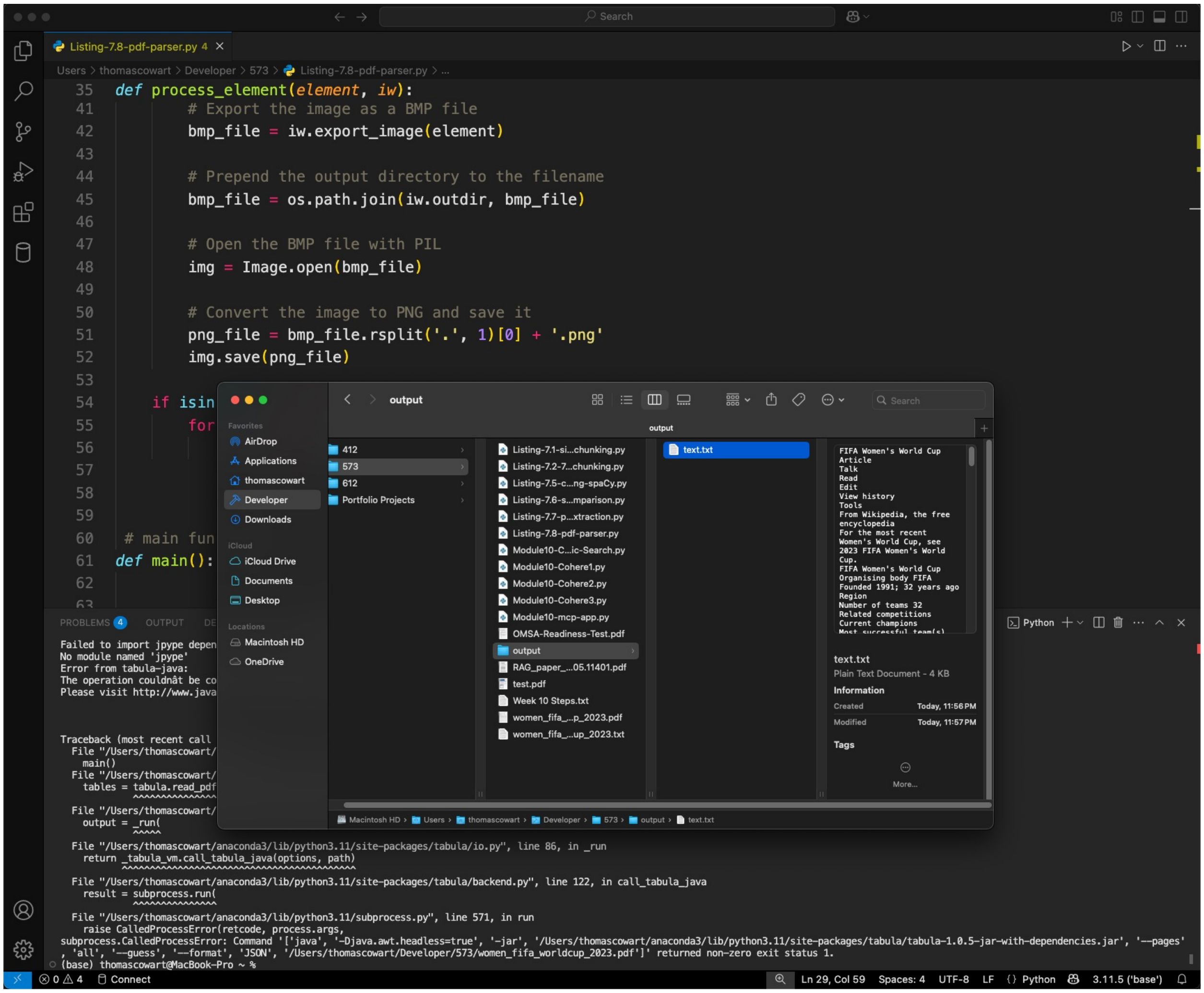
```python
def process_chunks(sentences):
        # Append the sentence and its embedding to the 2D array
        embedding = get_embedding(sentence)
        sentence_embeddings.append([sentence, embedding])

    #print("Simple Sentence Chunking:")
    print("\tNumber of sentence embeddings:", len(sentence_embeddings))
    print("\tTotal number of tokens:", total_token_count)

    return sentence_embeddings

if __name__ == "__main__":
    PDF_PATH = "/Users/thomascowart/Developer/573/women_fifa_worldcup_2023.pdf"
    extracted_text = extract_text_from_pdf(PDF_PATH)

    # Assuming a chunk size of 2000 characters for demonstration purposes.
    #chunks = paragraph_split(extracted_text, 2000)
    chunks = split_sentences_by_spacy(extracted_text, 2000)

    for index, chunk in enumerate(chunks):
        print(f"--- Chunk {index + 1} ---")
        print(chunk)
        print("----------------\n")

```

```
(chosen by the technical study group, awarded in 2019).
 World Cup Golden Ball Golden Boot Goals Golden Glove Clean sheets FIFA Young Player
Award FIFA Fair Play Trophy
China 1991 China United States Carin Jennings United States Michelle Akers 10 Not
Awarded N/A Not Awarded  Germany
Sweden 1995 Sweden Norway Hege Riise Norway Ann Kristin Aarønes 6  Sweden
United States 1999 United States China Sun Wen China Sun Wen
Brazil Sissi 7 China Gao Hong
United States Briana Scurry 5  China
United States 2003 United States Germany Birgit Prinz Germany Birgit Prinz 7 Germany
Silke Rottenberg 5  China
China 2007
---------------

--- Chunk 4 ---
China Brazil Marta Brazil Marta 7 Germany Nadine Angerer 6  Norway
Germany 2011 Germany Japan Homare Sawa Japan Homare Sawa 5 United States Hope Solo 2
Australia Caitlin Foord  Japan
Canada 2015 Canada United States Carli Lloyd Germany Célia Šašić 6 United States Hope Solo
5 Canada Kadeisha Buchanan  France
France 2019 France United States Megan Rapinoe United States Megan Rapinoe 6 Netherlands Sari
van Veenendaal 3 Germany Giulia Gwinn  France
AustraliaNew Zealand 2023 Australia/New Zealand Spain Aitana Bonmatí Japan Hinata Miyazawa
5 England Mary Earps 3 Spain Salma Paralluelo  Japan
---------------

(base) thomascowart@MacBook-Pro ~ % 
```

PDF chunking is about extracting text from PDFs, which often have complex layouts (like columns or images). It requires special tools to handle this extraction and organize the content.

Other chunking methods (like simple, textwrap, NLTK, and spaCy) work with plain text, splitting it into smaller pieces (sentences, words) without worrying about formatting or extraction.

9.



```
def process_element(element, iw):
        # Export the image as a BMP file
        bmp_file = iw.export_image(element)

        # Prepend the output directory to the filename
        bmp_file = os.path.join(iw.outdir, bmp_file)

        # Open the BMP file with PIL
        img = Image.open(bmp_file)

        # Convert the image to PNG and save it
        png_file = bmp_file.rsplit('.', 1)[0] + '.png'
        img.save(png_file)

    if isin
        for




# main fun
def main():
```

```
FIFA Women's World Cup
Article
Talk
Read
Edit
View history
Tools
From Wikipedia, the free
encyclopedia
For the most recent
Women's World Cup, see
2023 FIFA Women's World
Cup.
FIFA Women's World Cup
Organising body FIFA
Founded 1991; 32 years ago
Region
Number of teams 32
Related competitions
Current champions
Most successful team(s)
```

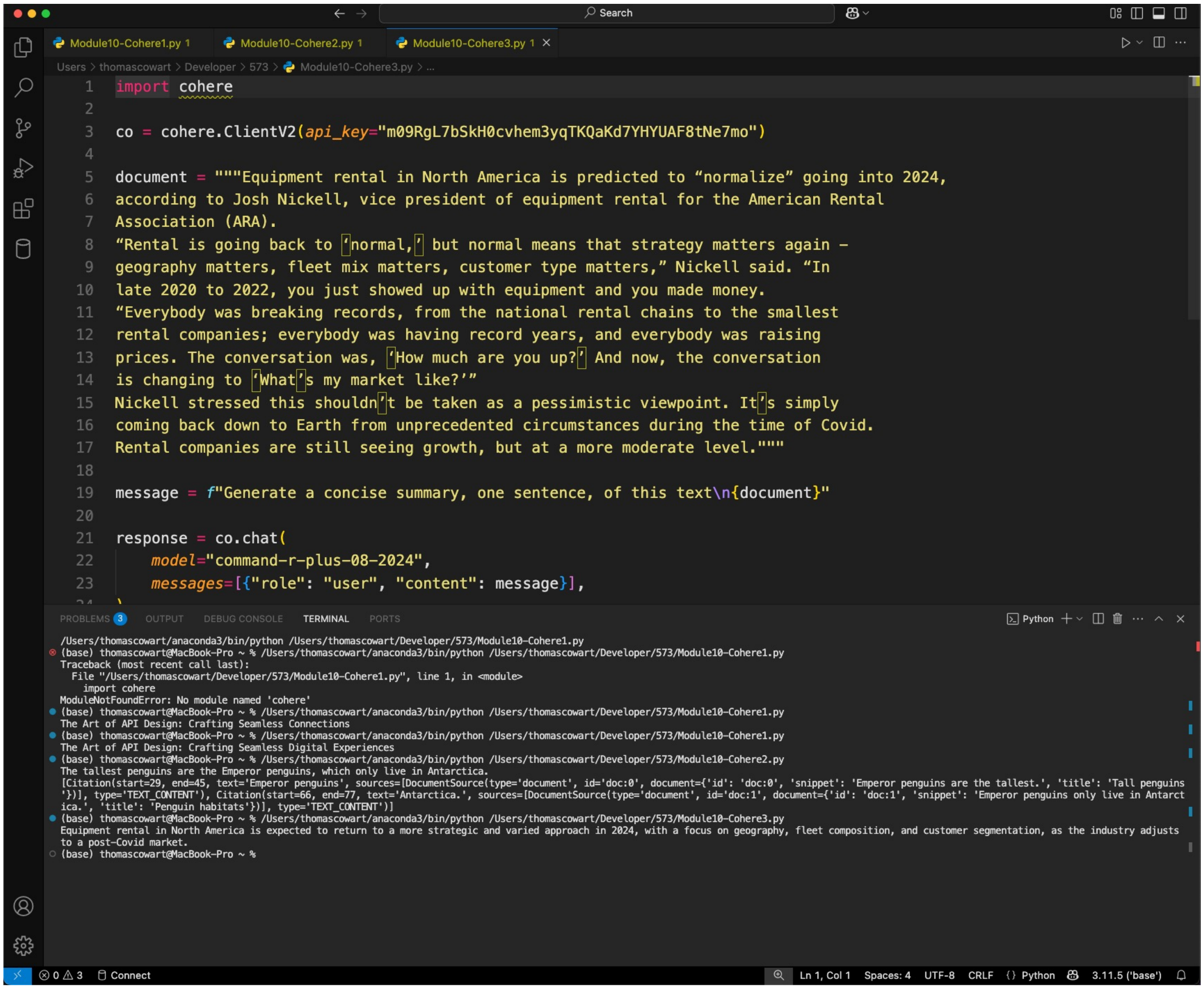
text.txt
Plain Text Document - 4 KB
Information
Created	Today, 11:56 PM
Modified	Today, 11:57 PM

Tags

```
Failed to import jpype depen
No module named 'jpype'
Error from tabula-java:
The operation couldnât be co
Please visit http://www.java


Traceback (most recent call
  File "/Users/thomascowart/
    main()
  File "/Users/thomascowart/
    tables = tabula.read_pdf
             ^^^^^^^^^^^^^^^
  File "/Users/thomascowart/
    output = _run(
             ^^^^^
  File "/Users/thomascowart/anaconda3/lib/python3.11/site-packages/tabula/io.py", line 86, in _run
    return _tabula_vm.call_tabula_java(options, path)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/Users/thomascowart/anaconda3/lib/python3.11/site-packages/tabula/backend.py", line 122, in call_tabula_java
    result = subprocess.run(
             ^^^^^^^^^^^^^^^
  File "/Users/thomascowart/anaconda3/lib/python3.11/subprocess.py", line 571, in run
    raise CalledProcessError(retcode, process.args,
subprocess.CalledProcessError: Command '['java', '-Djava.awt.headless=true', '-jar', '/Users/thomascowart/anaconda3/lib/python3.11/site-packages/tabula/tabula-1.0.5-jar-with-dependencies.jar', '--pages'
, 'all', '--guess', '--format', 'JSON', '/Users/thomascowart/Developer/573/women_fifa_worldcup_2023.pdf']' returned non-zero exit status 1.
(base) thomascowart@MacBook-Pro ~ %
```

You can extract text, tables, images, links/URLs, annotations/comments, form data, metadata, text structure, and fonts/layout from a PDF using tools like PyPDF2, pdfminer, pdfplumber, tabula, and PyMuPDF.

10.



```python
import cohere

co = cohere.ClientV2(api_key="m09RgL7bSkH0cvhem3yqTKQaKd7YHYUAF8tNe7mo")

document = """Equipment rental in North America is predicted to “normalize” going into 2024,
according to Josh Nickell, vice president of equipment rental for the American Rental
Association (ARA).
“Rental is going back to ‘normal,’ but normal means that strategy matters again –
geography matters, fleet mix matters, customer type matters,” Nickell said. “In
late 2020 to 2022, you just showed up with equipment and you made money.
“Everybody was breaking records, from the national rental chains to the smallest
rental companies; everybody was having record years, and everybody was raising
prices. The conversation was, ‘How much are you up?’ And now, the conversation
is changing to ‘What’s my market like?’”
Nickell stressed this shouldn’t be taken as a pessimistic viewpoint. It’s simply
coming back down to Earth from unprecedented circumstances during the time of Covid.
Rental companies are still seeing growth, but at a more moderate level."""

message = f"Generate a concise summary, one sentence, of this text\n{document}"

response = co.chat(
    model="command-r-plus-08-2024",
    messages=[{"role": "user", "content": message}],
)
```

```
/Users/thomascowart/anaconda3/bin/python /Users/thomascowart/Developer/573/Module10-Cohere1.py
(base) thomascowart@MacBook-Pro ~ % /Users/thomascowart/anaconda3/bin/python /Users/thomascowart/Developer/573/Module10-Cohere1.py
Traceback (most recent call last):
  File "/Users/thomascowart/Developer/573/Module10-Cohere1.py", line 1, in <module>
    import cohere
ModuleNotFoundError: No module named 'cohere'
(base) thomascowart@MacBook-Pro ~ % /Users/thomascowart/anaconda3/bin/python /Users/thomascowart/Developer/573/Module10-Cohere1.py
The Art of API Design: Crafting Seamless Connections
(base) thomascowart@MacBook-Pro ~ % /Users/thomascowart/anaconda3/bin/python /Users/thomascowart/Developer/573/Module10-Cohere1.py
The Art of API Design: Crafting Seamless Digital Experiences
(base) thomascowart@MacBook-Pro ~ % /Users/thomascowart/anaconda3/bin/python /Users/thomascowart/Developer/573/Module10-Cohere2.py
The tallest penguins are the Emperor penguins, which only live in Antarctica.
[Citation(start=29, end=45, text='Emperor penguins', sources=[DocumentSource(type='document', id='doc:0', document={'id': 'doc:0', 'snippet': 'Emperor penguins are the tallest.', 'title': 'Tall penguins
'})], type='TEXT_CONTENT'), Citation(start=66, end=77, text='Antarctica.', sources=[DocumentSource(type='document', id='doc:1', document={'id': 'doc:1', 'snippet': 'Emperor penguins only live in Antarct
ica.', 'title': 'Penguin habitats'})], type='TEXT_CONTENT')]
(base) thomascowart@MacBook-Pro ~ % /Users/thomascowart/anaconda3/bin/python /Users/thomascowart/Developer/573/Module10-Cohere3.py
Equipment rental in North America is expected to return to a more strategic and varied approach in 2024, with a focus on geography, fleet composition, and customer segmentation, as the industry adjusts
to a post-Covid market.
(base) thomascowart@MacBook-Pro ~ %
```

These programs demonstrate three different Cohere capabilities:

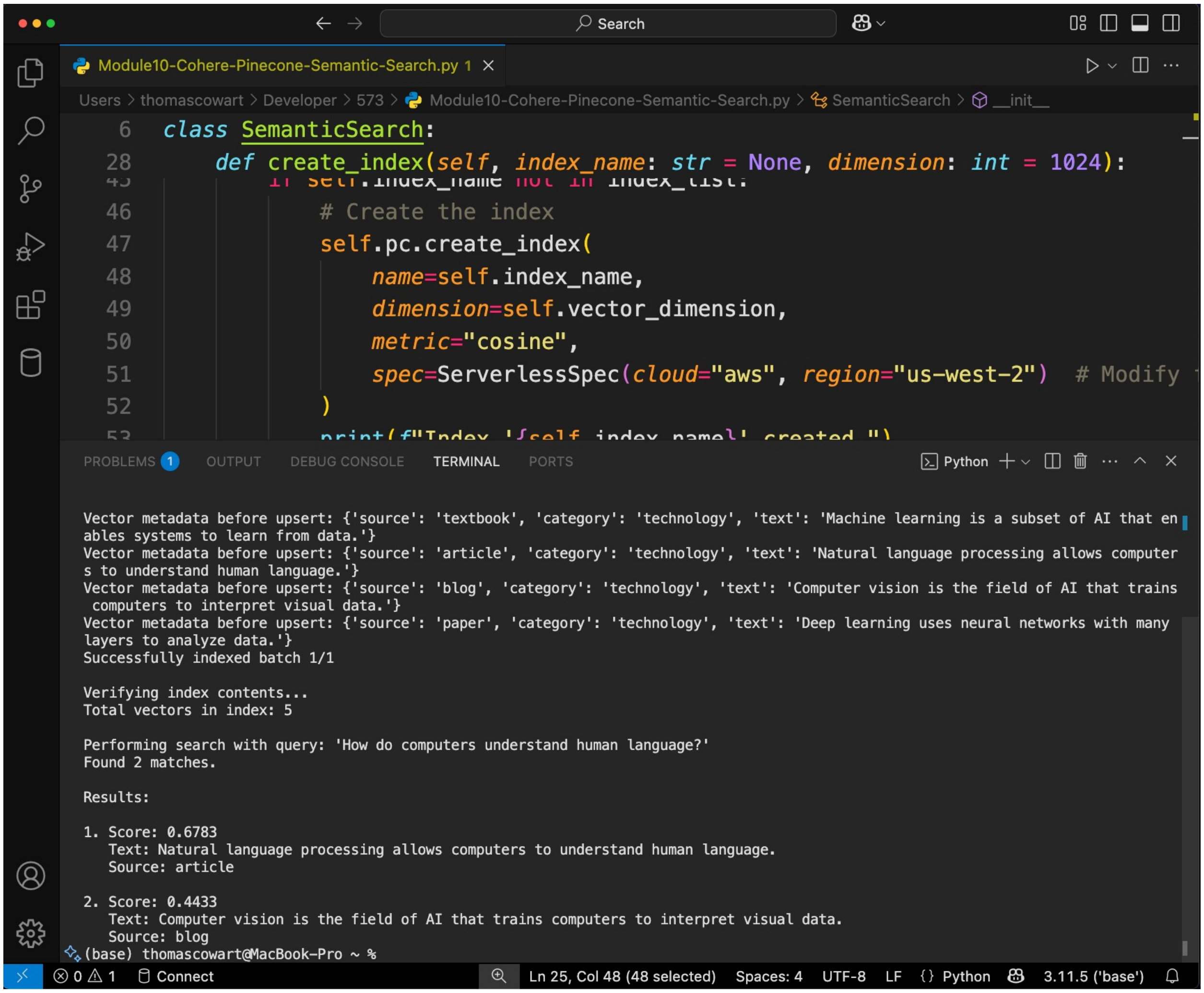
1. Chat-based API: In the first program, it uses the chat model to generate a title for a blog post.
2. Document Retrieval and Question Answering: The second program shows the ability to retrieve documents and use them to answer a user query by providing context from the documents.
3. Text Summarization: The third program generates a concise summary of a provided document.

Why use Cohere over OpenAI APIs?

- Customization: Cohere offers specialized models tailored for tasks like document retrieval and summarization.

- Efficiency: Cohere’s API might offer more cost-effective or optimized performance for specific use cases, especially in retrieving and processing documents.
- Model Specialization: Cohere’s models might be better for certain applications, such as context-based retrieval or summarization, depending on the task.

11.



Search
Module10-Cohere-Pinecone-Semantic-Search.py 1
Users > thomascowart > Developer > 573 > Module10-Cohere-Pinecone-Semantic-Search.py > SemanticSearch > __init__
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
0 1
Connect
Ln 25, Col 48 (48 selected)
Spaces: 4
UTF-8
LF
{} Python
3.11.5 ('base')

```python
class SemanticSearch:
    def create_index(self, index_name: str = None, dimension: int = 1024):
        if self.index_name not in index_list:
            # Create the index
            self.pc.create_index(
                name=self.index_name,
                dimension=self.vector_dimension,
                metric="cosine",
                spec=ServerlessSpec(cloud="aws", region="us-west-2")  # Modify
            )
            print(f"Index '{self.index_name}' created.")
```

```
Vector metadata before upsert: {'source': 'textbook', 'category': 'technology', 'text': 'Machine learning is a subset of AI that en
ables systems to learn from data.'}
Vector metadata before upsert: {'source': 'article', 'category': 'technology', 'text': 'Natural language processing allows computer
s to understand human language.'}
Vector metadata before upsert: {'source': 'blog', 'category': 'technology', 'text': 'Computer vision is the field of AI that trains
 computers to interpret visual data.'}
Vector metadata before upsert: {'source': 'paper', 'category': 'technology', 'text': 'Deep learning uses neural networks with many
layers to analyze data.'}
Successfully indexed batch 1/1

Verifying index contents...
Total vectors in index: 5

Performing search with query: 'How do computers understand human language?'
Found 2 matches.

Results:

1. Score: 0.6783
   Text: Natural language processing allows computers to understand human language.
   Source: article

2. Score: 0.4433
   Text: Computer vision is the field of AI that trains computers to interpret visual data.
   Source: blog
(base) thomascowart@MacBook-Pro ~ %
```

Using Pinecone with Cohere combines the power of fast, scalable vector search (Pinecone) with Cohere’s language models for enhanced AI capabilities. Pinecone stores and retrieves embeddings (vector representations of text), allowing you to quickly search through large datasets. When integrated with Cohere’s models, this enables tasks like semantic search, document retrieval, and personalized recommendations with high performance and relevance, making the combination ideal for real-time applications that require both powerful search and contextual understanding.