# Summarizer Task

## Generate a concise summary of the book "Crime and Punishment" limited to 20 pages

```
pip install pypdf
pip install fpdf
pip install langchain openai tiktoken fpdf2 pandas
pip install langchain_community
pip install chainlit
pip install faiss-cpu
pip install langchain_openai
pip install langchain_experimental
```

```
Collecting pypdf
  Downloading pypdf-5.1.0-py3-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: typing_extensions>=4.0 in /usr/local/lib/python3.10/dist-packages (from pypdf) (4.12.2)
Downloading pypdf-5.1.0-py3-none-any.whl (297 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 298.0/298.0 kB 4.7 MB/s eta 0:00:00
Installing collected packages: pypdf
Successfully installed pypdf-5.1.0
```

```
from langchain.document_loaders import PyPDFLoader

# Load the book
loader = PyPDFLoader("crime-and-punishment.pdf")
pages = loader.load_and_split()
```

```
page
```

```
# Cut out the open and closing parts
pages = pages[7:]
# Combine the pages, and replace the tabs with spaces
text = ' '.join([page.page_content.replace('\t', '') for page in pages])
```

```
text[0:1000]
```

```
'Crime and Punishment\x18\nstopped on the stairs, to be forced to listen to her trivial, ir -\nrelevant gossip, to pesterin
g demands for payment, threats \nand complaints, and to rack his brains for excuses, to pre -\nvaricate, to lie—no, rather
than that, he would creep down \nthe stairs like a cat and slip out unseen.\nThis evening, however, on coming out into the
street, he \nbecame acutely aware of his fears.\n‘I want to attempt a thing like that  and am frightened \nby these trifle
s,’ he thought, with an odd smile. ‘Hm … yes, \nall is in a man’s hands and he lets it all slip from cowardice, \nthat’s an
axiom. It would be interesting to know what it is \nmen are most afraid of. Taking a new step, uttering a new \nword is wha
t they fear most…. But I am talking too much. \nIt’s because I chatter that I do nothing. Or perhaps it is that \nI chatter
because I do nothing. I’ve learned to chatter this \nlast month, lying for days together in my den thinking … \nof Jack the
```

```
import re
def clean_text(text):
  cleaned_text = text

   # Remove extra spaces
  cleaned_text = re.sub(r' +', ' ', cleaned_text)
   # Replace newline characters with spaces
  cleaned_text = cleaned_text.replace('\n', ' ')
  return cleaned_text
clean_text=clean_text(text)
```

```
clean_text
```

```
'Crime and Punishment\x18 stopped on the stairs, to be forced to listen to her trivial, ir - relevant gossip, to pestering
demands for payment, threats  and complaints, and to rack his brains for excuses, to pre - varicate, to lie—no, rather than
that, he would creep down  the stairs like a cat and slip out unseen. This evening, however, on coming out into the street,
he  became acutely aware of his fears. ‘I want to attempt a thing like that and am frightened  by these trifles,’ he though
t, with an odd smile. ‘Hm … yes,  all is in a man’s hands and he lets it all slip from cowardice,  that’s an axiom. It woul
d be interesting to know what it is  men are most afraid of. Taking a new step, uttering a new  word is what they fear mos
t…. But I am talking too much.  It’s because I chatter that I do nothing. Or perhaps it is that  I chatter because I do not
hing. I’ve learned to chatter this  last month, lying for days together in my den thinking …  of Jack the Giant-killer. Why
```

```
import os
os.environ["OPENAI_API_KEY"] = "your API key"
```

```
from langchain import OpenAI
llm = OpenAI()
Tokens = llm.get_num_tokens(clean_text)
print (f"We have {Tokens} tokens in the book")
```

```
We have 295753 tokens in the book
```

```
from langchain_experimental.text_splitter import SemanticChunker
from langchain_openai.embeddings import OpenAIEmbeddings
text_splitter = SemanticChunker(
   OpenAIEmbeddings(), breakpoint_threshold_type="interquartile"
)
docs = text_splitter.create_documents([clean_text])
```

```
docs
```

```
Sonia had only been able to visit  him twice during his illness; each time she had to obtain  permission, and it was difficult. But she often used to come  to the hospital yard, especially in the evening, sometimes  only to stand a minute and look up at the windows of the  ward. One evening, when he was almost well again, Raskol - nikov fell asleep. On waking up he chanced to go to the  window, and at once saw Sonia in the distance at the hospi - tal gate. She seemed to be waiting for someone. Something  stabbed him to the heart at that minute. He shuddered and  moved away from the window. Next day Sonia did not come,  nor the day after; he noticed that he was expecting her un - easily. At last he was discharged. On reaching the prison he  learnt from the convicts that Sofya Semyonovna was lying  ill at home and was unable to go out. He was very uneasy and sent to inquire after her; he soon  learnt that her illness was not dangerous. Hearing that he  was anxious about her, Sonia sent him a pencilled note, tell - ing him that she was much better, that she had a slight cold and that she would soon, very soon come and see him at his  work. His heart throbbed painfully as he read it. Again it was a warm bright day. Early in the morning,  at six o'clock, he went off to work on the river bank, where  they used to pound alabaster and where there was a kiln  for baking it in a shed. There were only three of them sent. One of the convicts went with the guard to the fortress to Crime and Punishment\x18\x18\x18 fetch a tool; the other began getting the wood ready and laying it in the kiln. Raskolnikov came out of the shed on  to the river bank, sat down on a heap of logs by the shed  and began gazing at the wide deserted river. From the high  bank a broad landscape opened before him, the sound of  singing floated faintly audible from the other bank. In the  vast steppe, bathed in sunshine, he could just see, like black specks, the nomads' tents. There there was freedom, there  other men were living, utterly unlike those here; there time itself seemed to stand still, as though the age of Abraham  and his flocks had not passed. Raskolnikov sat gazing, his thoughts passed into day-dreams, into contemplation; he  thought of nothing, but a vague restlessness excited and  troubled him. Suddenly he found Sonia beside him; she  had come up noiselessly and sat down at his side. It was  still quite early; the morning chill was still keen. She wore  her poor old burnous and the green shawl; her face still  showed signs of illness, it was thinner and paler. She gave  him a joyful smile of welcome, but held out her hand with  her usual timidity. She was always timid of holding out  her hand to him and sometimes did not offer it at all, as  though afraid he would repel it. He always took her hand  as though with repugnance, always seemed vexed to meet  her and was sometimes obstinately silent throughout her  visit. Sometimes she trembled before him and went away  deeply grieved. But now their hands did not part. He stole  a rapid glance at her and dropped his eyes on the ground  without speaking. They were alone, no one had seen them. The guard had turned away for the time. \x18\x18\x18Free eBooks at Planet eBook.com How it happened he did not know. But all at once some - thing seemed to seize him and fling him at her feet. He wept  and threw his arms round her knees. For the first instant  she was terribly frightened and she turned pale. She jumped  up and looked at him trembling. But at the same moment  she understood, and a light of infinite happiness came into  her eyes. She knew and had no doubt that he loved her be - yond everything and that at last the moment had come…. They wanted to speak, but could not; tears stood in their  eyes. They were both pale and thin; but those sick pale faces  were bright with the dawn of a new future, of a full res - urrection into a new life. They were renewed by love; the  heart of each held infinite sources of life for the heart of the  other. They resolved to wait and be patient. They had another  seven years to wait, and what terrible suffering and what  infinite happiness before them! But he had risen again and  he knew it and felt it in all his being, while she—she only  lived in his life. On the evening of the same day, when the barracks were  locked, Raskolnikov lay on his plank bed and thought of  her. He had even fancied that day that all the convicts who  had been his enemies looked at him differently; he had  even entered into talk with them and they answered him  in a friendly way. He remembered that now, and thought  it was bound to be so. Wasn't everything now bound to be  changed? He thought of her. He remembered how continually he  had tormented her and wounded her heart. He remembered Crime and Punishment\x18\x18\x18 her pale and thin little face. But these recollections scarcely  troubled him now; he knew with what infinite love he would  now repay all her sufferings. And what were all, all the ago- nies of the past! Everything, even his crime, his sentence  and imprisonment, seemed to him now in the first rush of  feeling an external, strange fact with which he had no con - cern. But he could not think for long together of anything  that evening, and he could not have analysed anything con - sciously; he was simply feeling. Life had stepped into the  place of theory and something quite different would work  itself out in his mind. Under his pillow lay the New Testament. He took it up  mechanically. The book belonged to Sonia; it was the one from which she had read the raising of Lazarus to him. At  first he was afraid that she would worry him about religion, would talk about the gospel and pester him with books. But  to his great surprise she had not once approached the sub - ject and had not even offered him the Testament. He had  asked her for it himself not long before his illness and she brought him the book without a word. Till now he had not  opened it. He did not open it now, but one thought passed through his mind: 'Can her convictions not be mine now? Her feel - ings, her aspirations at least….' She too had been greatly agitated that day, and at night  she was taken ill again. But she was so happy—and so un - expectedly happy—that she was almost frightened of her  happiness. Seven years, only seven years! At the beginning  of their happiness at some moments they were both ready \x18\x18\x18Free eBooks at Planet eBook.com to look on those seven years as though they were seven days. He did not know that the new life would not be given him  for nothing, that he would have to pay dearly for it, that it  would cost him great striving, great suffering. But that is the beginning of a new story—the story of the  gradual renewal of a man, the story of his gradual regen - eration, of his passing from one world into another, of his  initiation into a new unknown life. That might be the sub - ject of a new story, but our present story is ended.')]
```

```
import numpy as np
import openai
def get_embeddings(text):
```

```
    response = openai.embeddings.create(
        model="text-embedding-3-small",
        input=text
    )
    return response.data
embeddings=get_embeddings([doc.page_content for doc in docs]
)


import pandas as pd
content_list = [doc.page_content for doc in docs]
df = pd.DataFrame(content_list, columns=['page_content'])
vectors = [embedding.embedding for embedding in embeddings]
array = np.array(vectors)
embeddings_series = pd.Series(list(array))
df['embeddings'] = embeddings_series


import numpy as np
import faiss
# Convert to float32 if not
array = array.astype(np.float32)
num_clusters = 50
# Vectors dimensionality
dimension = array.shape[1]
# Train KMeans with Faiss
kmeans = faiss.Kmeans(dimension, num_clusters, niter=20, verbose=True)
kmeans.train(array)
# Directly access the centroids
centroids = kmeans.centroids
# Create a new index for the original dataset
index = faiss.IndexFlatL2(dimension)
# Add original dataset to the index
index.add(array)


dimension
```

```
1536
```

```
centroids
```

```
array([[-0.00555586,  0.01042398, -0.02900542, ...,  0.00193562,
         0.00738207,  0.00542081],
       [ 0.00393355, -0.00260278, -0.05338135, ...,  0.00586344,
        -0.00142796, -0.00865541],
       [-0.0042717 ,  0.02227477, -0.06461308, ..., -0.00597494,
         0.00159662,  0.00590726],
       ...,
       [ 0.00509555,  0.01457771, -0.03853327, ...,  0.00764914,
        -0.00579194,  0.01315463],
       [ 0.00689662, -0.00264981, -0.03400137, ...,  0.00484546,
        -0.00364802, -0.01404972],
       [ 0.01037996,  0.04096801, -0.09780817, ...,  0.01972193,
         0.01776419,  0.00986097]], dtype=float32)
```

```
centroids.shape
```

```
(50, 1536)
```

```
D, I = index.search(centroids, 1)


print(D, I)
```

```
[[5.5531108e-01]
 [2.3152131e-01]
 [2.5705433e-01]
 [5.2105987e-01]
 [2.1034080e-01]
 [2.2913629e-01]
 [3.6604929e-01]
 [3.0525213e-01]
 [0.0000000e+00]
 [2.6197076e-01]
 [1.9339812e-01]
 [4.3454808e-01]
 [2.4064034e-01]
 [3.3409023e-01]
```

```
       [0.0000000e+00]
       [4.7683716e-07]
       [4.7273737e-01]
       [3.5261226e-01]
       [3.9255464e-01]
       [3.3961195e-01]
       [2.1463037e-01]
       [1.9713461e-01]
       [2.2124457e-01]
       [4.5171750e-01]
       [1.9406152e-01]
       [2.1862435e-01]
       [2.6064688e-01]
       [1.6917598e-01]
       [2.3841858e-07]
       [2.7727890e-01]
       [1.3234866e-01]
       [1.4288169e-01]
       [1.3524222e-01]
       [2.4872470e-01]
       [3.7589276e-01]
       [4.3976182e-01]
       [3.7593102e-01]
       [3.5214096e-01]
       [1.3747036e-01]
       [3.2897955e-01]
       [1.4859200e-01]
       [1.7357135e-01]
       [2.3576701e-01]
       [3.4277731e-01]
       [4.7683716e-07]
       [3.3010232e-01]
       [9.5367432e-07]
       [5.4962742e-01]
       [3.5536349e-01]
       [0.0000000e+00]] [[464]
       [191]
       [379]
       [118]
       [258]
       [ 89]
       [418]
       [ 61]
       [441]
```

```
sorted_array = np.sort(I, axis=0)
sorted_array=sorted_array.flatten()
extracted_docs = [docs[i] for i in sorted_array]
```

```
from langchain_openai import ChatOpenAI

# Instantiating the model
model = ChatOpenAI(temperature=0, model="gpt-4")
```

```
from langchain_core.output_parsers import StrOutputParser
from langchain_openai import ChatOpenAI
from langchain_core.prompts import ChatPromptTemplate
prompt = ChatPromptTemplate.from_template("""
"Provide a clear and concise overview of the given PDF, capturing all essential points, themes, and highlights while maintaining
"Exclude minor details and repetitive information, diving directly into the narrative or descriptions from the text.\n"
"Address the main events, characters, and themes in a detailed and engaging manner, ensuring the summary encapsulates the essenc
"Present a unified and coherent narrative that feels like a natural progression of the original text, ensuring readability and l

Passage:

```{text}```
SUMMARY:
"""
)
```

```
chain= (
    prompt
   | model
   |StrOutputParser() )
```

```
from tqdm import tqdm
final_summary = ""
```
```

```
for doc in tqdm(extracted_docs, desc="Processing documents"):
   # Get the new summary.
   new_summary = chain.invoke({"text": doc.page_content})
   # Update the list of the last two summaries: remove the first one and add the new one at the end.
   final_summary+=new_summary
```

```
Processing documents: 100%|██████████| 50/50 [08:15<00:00,  9.91s/it]
```

```
from fpdf import FPDF

class PDF(FPDF):
   def header(self):
       # Select Arial bold 15
       self.set_font('Arial', 'B', 15)
       # Move to the right
       self.cell(80)
       # Framed title
       self.cell(30, 10, 'Summary', 1, 0, 'C')
       # Line break
       self.ln(20)

   def footer(self):
       # Go to 1.5 cm from bottom
       self.set_y(-15)
       # Select Arial italic 8
       self.set_font('Arial', 'I', 8)
       # Page number
       self.cell(0, 20, 'Page %s' % self.page_no(), 0, 0, 'C')

# Instantiate PDF object and add a page
pdf = PDF()
pdf.add_page()
pdf.set_font("Arial", size=12)

# Ensure the 'final_summary' text is treated as UTF-8

# Make sure your text is a utf-8 encoded string
last_summary_utf8 = final_summary.encode('latin-1', 'replace').decode('latin-1')
pdf.multi_cell(0, 10, last_summary_utf8)

# Save the PDF to a file
pdf_output_path = "summ_output.pdf"
pdf.output(pdf_output_path)
```

```
''
```

# Task 2

## Personalized Study Plans For Students Using LangChain

```
pip install langchain_openai
```

Show hidden output

```
from langchain.prompts import PromptTemplate
from langchain.chat_models import ChatOpenAI

# Initialize OpenAI API
import os
os.environ["OPENAI_API_KEY"] = "your API key"
llm = ChatOpenAI(model="gpt-4", temperature=0.3)

# Define the detailed prompt template
prompt_template = PromptTemplate(
    input_variables=[
        "name", "field_of_study", "year_of_study", "list_of_subjects", "preferred_learning_styles",
        "personal_objectives", "challenges", "extracurricular_activities"
    ],
    template=(
        "You are an expert academic planner. Your task is to create a personalized study plan for a student based on the followin
        "Student Name: {name}\n"
```

```
            "Field of Study: {field_of_study}\n"
            "Year of Study: {year_of_study}\n"
            "List of Subjects: {list_of_subjects} (e.g., Math, Physics, Literature)\n"
            "Preferred Learning Styles: {preferred_learning_styles} (e.g., visual, auditory, kinesthetic)\n"
            "Personal Objectives: {personal_objectives} (e.g., preparing for exams, improving specific subjects)\n"
            "Challenges: {challenges} (e.g., time management issues, difficulty concentrating)\n"
            "Extracurricular Activities: {extracurricular_activities} (e.g., sports, music, coding)\n\n"
            "Using this information, create a study plan that includes:\n"
            "1. Weekly schedules customized to the student's learning style and academic requirements.\n"
            "2. Strategies to address the challenges mentioned.\n"
            "3. Suggestions for balancing extracurricular activities with academic goals.\n"
            "4. Specific action items for achieving personal objectives.\n"
            "5. Resources or tools that align with the student's learning style and needs.\n\n"
            "Ensure the plan is detailed, actionable, and motivating for the student."
        )
    )

    # Example usage
    example_input = {
        "name": "Alex Johnson",
        "field_of_study": "Computer Science",
        "year_of_study": "2nd Year",
        "list_of_subjects": "Data Structures, Algorithms, Operating Systems",
        "preferred_learning_styles": "Visual and kinesthetic",
        "personal_objectives": "Excel in Data Structures and prepare for internships",
        "challenges": "Balancing coursework with personal projects",
        "extracurricular_activities": "Robotics Club, Basketball"
    }

    # Render the prompt with example data
    filled_prompt = prompt_template.format(**example_input)


    response = llm.predict(filled_prompt)

    # Output the result
    print(response)
```

```
Monday:
- 9am-11am: Data Structures (Visual learning: Use diagrams and flowcharts)
- 11am-12pm: Break (Basketball practice)
- 12pm-2pm: Algorithms (Kinesthetic learning: Hands-on coding exercises)
- 2pm-3pm: Lunch
- 3pm-5pm: Personal Project Time

Tuesday:
- 9am-11am: Operating Systems (Visual learning: Watch video tutorials)
- 11am-12pm: Break (Robotics Club)
- 12pm-2pm: Algorithms (Kinesthetic learning: Solve algorithm problems)
- 2pm-3pm: Lunch
- 3pm-5pm: Data Structures (Visual learning: Review diagrams and flowcharts)

Wednesday:
- 9am-11am: Data Structures (Kinesthetic learning: Implement data structures)
- 11am-12pm: Break (Basketball practice)
- 12pm-2pm: Operating Systems (Visual learning: Review video tutorials)
- 2pm-3pm: Lunch
- 3pm-5pm: Personal Project Time

Thursday:
- 9am-11am: Algorithms (Visual learning: Review algorithm diagrams)
- 11am-12pm: Break (Robotics Club)
- 12pm-2pm: Data Structures (Kinesthetic learning: Hands-on coding exercises)
- 2pm-3pm: Lunch
- 3pm-5pm: Operating Systems (Kinesthetic learning: Hands-on OS tasks)

Friday:
- 9am-11am: Review of the Week's Work
- 11am-12pm: Break (Basketball practice)
- 12pm-2pm: Internship Preparation (Resume building, interview practice)
- 2pm-3pm: Lunch
- 3pm-5pm: Personal Project Time

2. Strategies to Address Challenges:
```

```
- Schedule your extracurricular activities during your break times or after your study sessions. This will ensure they don't
- Use your involvement in the Robotics Club to enhance your learning. Try to apply what you learn in your computer science c
- Basketball can be a great way to relieve stress and stay active. Try to incorporate it into your daily routine as a form o

4. Action Items for Personal Objectives:

- Dedicate at least two hours each day to studying Data Structures. Use visual and kinesthetic learning techniques to enhanc
- Spend at least two hours each week preparing for internships. This could involve improving your resume, practicing coding
- Use your personal project time to work on projects that will showcase your skills to potential employers.

5. Resources:

- Visual learners: Use resources like YouTube, Coursera, and Khan Academy for video tutorials.
- Kinesthetic learners: Use platforms like LeetCode, HackerRank, and Codecademy for hands-on coding practice.
- For internship preparation, use resources like Cracking the Coding Interview, LeetCode, and Pramp for interview practice.
```

Start coding or generate with AI.