```
import os
from PyPDF2 import PdfReader
import pdfplumber
from sentence_transformers import SentenceTransformer
import pinecone

# Initialize the embedding model
embedding_model = SentenceTransformer('all-MiniLM-L6-v2')

# Initialize the Pinecone vector database
pinecone.init(api_key="<sk-proj-tskVYFwmETi2Y5LBUzYOEUG3aNRMDk7mPbFGBkhDzshAAakA9od-
1II5A7mohQV4S8Lfxe3hjeT3BlbkFJ4Iynv04e4ddXEwZ0kQL7qlXF-Qnh-
9gvats22RlSPU0zLmjB5wPIW_J2b9F7eElpMB0ebVAesA>", environment="us-west1-gcp")
index = pinecone.Index("rag-pipeline-index")

# Function to extract text from PDF using pdfplumber
def extract_text_from_pdf(pdf_path):
    with pdfplumber.open(pdf_path) as pdf:
        text_data = []
        for page in pdf.pages:
            text_data.append(page.extract_text())
    return text_data

# Function to chunk text into smaller segments
def chunk_text(text, max_chunk_size=300):
    words = text.split()
    chunks = []
    current_chunk = []

    for word in words:
        current_chunk.append(word)
        if len(current_chunk) >= max_chunk_size:
            chunks.append(" ".join(current_chunk))
            current_chunk = []

    if current_chunk:
        chunks.append(" ".join(current_chunk))

    return chunks

# Function to embed and store chunks in Pinecone
def store_chunks_in_pinecone(chunks, metadata):
    for chunk in chunks:
        embedding = embedding_model.encode(chunk).tolist()
        index.upsert([(metadata['id'], embedding, metadata)])
```

```
# Main pipeline function
def process_pdf(pdf_path):
    # Extract text from the PDF
    text_data = extract_text_from_pdf(pdf_path)

    for page_num, page_text in enumerate(text_data):
        # Chunk text into smaller pieces
        chunks = chunk_text(page_text)

        # Metadata for the page
        metadata = {
            "id": f"{os.path.basename(pdf_path)}_page_{page_num}",
            "file_name": os.path.basename(pdf_path),
            "page_number": page_num
        }

        # Store chunks in Pinecone
        store_chunks_in_pinecone(chunks, metadata)

# Example usage
if __name__ == "__main__":
    pdf_file_path = "example.pdf"  # Path to your PDF file
    process_pdf(pdf_file_path)

    # Example query processing
    query = "What is the unemployment rate for those with a bachelor's degree?"
    query_embedding = embedding_model.encode(query).tolist()

    # Perform similarity search in Pinecone
    results = index.query(query_embedding, top_k=5, include_metadata=True)

    for match in results["matches"]:
        print(f"Page {match['metadata']['page_number']}: {match['metadata']}")
```