

AI AGENT For a SWOT Analysis

Created by:

Parikshith M

AI Agent: Swot Analysis Generator

- Model used: "microsoft/Phi-2" a Hugging Face Model
- The Agent takes 2 modes of input:
 - The text box
 - File uploader in the format of PDF, DOCX, TXT and CSV formats.

Code used to build the agent:

```
#!/usr/bin/env python
```

```
# coding: utf-8
```

```
import os
```

```
import asyncio
```

```
import nest_asyncio
```

```
import streamlit as st
```

```
import torch
```

```
from transformers import pipeline, AutoModelForCausalLM, AutoTokenizer
```

```
from PyPDF2 import PdfReader
```

```
import docx
```

Print Library Versions

```
print("Library Versions:")
```

```
print(f"Streamlit: {st.__version__}")
```

```
print(f"Transformers: {pipeline.__module__.split('.')[0]}")
```

```
print(f"PyTorch: {torch.__version__}")
```

```
print(f"PyPDF2: {PdfReader.__module__.split('.')[0]}")
```

```
print(f"python-docx: {docx.__version__ if hasattr(docx, '__version__') else 'Unknown'}")
```

Apply nest_asyncio to handle event loop issues

```
nest_asyncio.apply()
```

Set up Hugging Face Microsoft model

```
MODEL_NAME = "microsoft/Phi-2" # Example Microsoft model; change if needed
```

Load tokenizer and model

```
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
```

```
model = AutoModelForCausalLM.from_pretrained(
```

```
    MODEL_NAME,
```

```
    torch_dtype=torch.float16,
```

```
    device_map="auto"
```

```
)
```

Load text generation pipeline

```
text_generator = pipeline("text-generation", model=model, tokenizer=tokenizer)
```

Initialize session states for token tracking

```
if "tokens_consumed" not in st.session_state:
```

```
    st.session_state.tokens_consumed = 0
```

Define the prompt template for SWOT Analysis

```
prompt_template = """
```

```
You are an AI assistant specializing in business analysis. Given the following organization details,  
generate a SWOT analysis.
```

```
Provide:
```

```
- Strengths
```

```
- Weaknesses
```

```
- Opportunities
```

```
- Threats
```

```
Organization Details:
```

```
{context}
```

SWOT Analysis:

```
"""
```

```
def extract_text_from_pdf(pdf_file):
```

```
    """Extract text from a PDF file."""
```

```
    pdf_reader = PdfReader(pdf_file)
```

```
    text = ""
```

```
    for page in pdf_reader.pages:
```

```
        text += page.extract_text() + "\n"
```

```
    return text
```

```
def extract_text_from_docx(docx_file):
```

```
    """Extract text from a DOCX file."""
```

```
    doc = docx.Document(docx_file)
```

```
    text = "\n".join([para.text for para in doc.paragraphs])
```

```
    return text
```

```
def generate_swot_analysis(txt):
```

```
    """Generate a SWOT analysis from input text."""
```

```
    prompt = prompt_template.format(context=txt)
```

```
# Generate text using Hugging Face model
```

```
    output = text_generator(prompt, max_length=5000, do_sample=True, temperature=0.7)
```

```
    return output[0]["generated_text"]
```

```
# Streamlit UI
```

```
st.set_page_config(page_title="SWOT Analysis App")
```

```
st.title("SWOT Analysis Generator")
```

```
st.write("Upload a file or enter details about an organization to generate a SWOT analysis.")
```

File Upload Section

```
uploaded_file = st.file_uploader("Upload a PDF, DOCX, TXT, or CSV file:", type=["pdf", "docx", "txt", "csv"])
```

Text Input Section

```
text_input = st.text_area("Or, enter organization details manually:")
```

Read file content if uploaded

```
file_text = ""
```

```
if uploaded_file:
```

```
    file_type = uploaded_file.type
```

```
    if file_type == "text/plain": # TXT file
```

```
        file_text = uploaded_file.read().decode("utf-8")
```

```
    elif file_type == "application/pdf": # PDF file
```

```
        file_text = extract_text_from_pdf(uploaded_file)
```

```
    elif file_type == "application/vnd.openxmlformats-officedocument.wordprocessingml.document": # DOCX file
```

```
        file_text = extract_text_from_docx(uploaded_file)
```

```
    elif file_type == "text/csv": # CSV file
```

```
        file_text = uploaded_file.read().decode("utf-8")
```

Final input text (priority: file > manual input)

```
final_text = file_text.strip() if file_text else text_input.strip()
```

```
if final_text:
```

```
    st.subheader("SWOT Analysis:")
```

```
    with st.spinner("Generating SWOT analysis..."):
```

```
        swot_analysis = generate_swot_analysis(final_text)
```

```
    st.write(swot_analysis)
```

Token count update (approximate)

```
st.session_state.tokens_consumed += len(tokenizer.encode(final_text))
```

Sidebar token tracking

```
st.sidebar.write(f"Total Tokens Consumed: {st.session_state.tokens_consumed}")
```

```
print("Tokens consumed in this transaction:", st.session_state.tokens_consumed)
```

Reset token count after transaction

```
st.session_state.tokens_consumed = 0
```

The AI AGENT Streamlit Window

SWOT Analysis Generator

Upload a file or enter details about an organization to generate a SWOT analysis.

Upload a PDF, DOCX, TXT, or CSV file:



Drag and drop file here

Limit 200MB per file • PDF, DOCX, TXT, CSV

Browse files



NovaEdge Industries.docx 14.9KB



Or, enter organization details manually:

INPUT To the AI Agent: The Nova Edge Company details Provided.

Output from the AI Agent:

SWOT Analysis: Strengths:

- Proactive investment in digital transformation
- Committed to flexible workforce policies
- Ability to adapt to changing market conditions
- Strong foundation for future success

Weaknesses:

- Reliance on legacy systems
- Challenges with communication hurdles
- Limited research and development capabilities compared to larger competitors
- Competition from technology-forward rivals

Opportunities:

- Expanding into new international markets
- Forming strategic partnerships with tech startups and academic institutions
- Mitigating risks associated with geopolitical instability and regulatory changes
- Broadening supplier base

Threats:

- Competition from technology-forward rivals
- Aggressive market strategies of competitors
- Set new benchmarks for efficiency and innovation
- Communication challenges within the organization
- Reliance on legacy systems
- Intensified competition for top talent

The total Tokens consumed:

Total Tokens Consumed: 688