

Creación de un Chatbot con Personalidad de Javier Santaolalla

5 de Septiembre

Resumen

Este documento explica paso a paso el proceso de creación e implementación de un chatbot con la personalidad del físico y divulgador Javier Santaolalla. El chatbot fue desarrollado en Python utilizando Streamlit, LangChain, DeepSeek y gTTS, permitiendo la interacción tanto por texto como por voz. Finalmente, se despliega en la nube mediante Streamlit Cloud.

-

Índice

1. Introducción

2. Herramientas Utilizadas

3. Proceso de Creación

3.1 Creación del entorno de trabajo

3.2 Instalación de dependencias

3.3 Creación de los archivos principales

3.4 Ejecución local

3.5 Control de versiones con GitHub

3.6 Despliegue en Hugging face

4. Desarrollo del Chatbot

4.1 Definición de la personalidad

4.2 Interacción con el usuario

4.3 Procesamiento de la respuesta

4.4 Presentación de la respuesta

5. Conclusión

1. Introducción

El objetivo de este proyecto fue crear un chatbot educativo que pueda responder preguntas relacionadas con **sistemas digitales**, con un estilo de comunicación inspirado en Javier Santaolalla.

2. Herramientas Utilizadas

- Python 3.12
- Streamlit
- streamlit-mic-recorder
- LangChain y langchain-deepseek
- gTTS (Google Text-to-Speech)
- GitHub
- Streamlit Cloud o Hugging face

3. Proceso de Creación

3.1. Creación del entorno de trabajo

1. Crear carpeta: `mkdir chatbot_pepper`
2. Entrar en carpeta: `cd chatbot_pepper`
3. Crear entorno virtual: `python3 -m venv venv`
4. Activar entorno virtual:
 - Linux/Mac: `source venv/bin/activate`

3.2. Instalación de dependencias

```
pip install streamlit streamlit-mic-recorder gtts
pip install langchain langchain-deepseek
pip install speechrecognition pydub
```

3.3. Creación de archivos principales

- `chatbot.py` con el código del chatbot.
- `requirements.txt` con dependencias.

3.4. Ejecución local

```
streamlit run chatbot.py
```

3.5. Control de versiones con GitHub

```
git init
git add .
git commit -m "Primer commit"
git remote add origin <URL_REPOSITORIO>
git push origin main
```

3.6. Despliegue en Huggin Face

1. Conectar GitHub con Huggin Face.
2. Subir `requirements.txt` y `chatbot.py`
3. Guardar API Key de DeepSeek en “Secrets”.
4. Poner sdk : streamlit en el readme (para ejecucion con stremlit)
5. Ejecutar la app en huggin face

4. Desarrollo del Chatbot

4.1. Definición de la personalidad

El chatbot adopta la personalidad de Javier Santaolalla mediante un prompt definido en el código:

“Eres un chatbot con la personalidad de Javier Santaolalla, físico y divulgador científico. Debes comunicarte como él: apasionado por la ciencia, cercano, entusiasta y con un toque de humor...”

4.2. Interacción con el usuario

El usuario puede:

- Escribir texto mediante `st.chat_input`.
- Hablar al micrófono mediante `streamlit-mic-recorder`.

4.3. Procesamiento de la respuesta

- DeepSeek genera la respuesta en texto.
- gTTS convierte la respuesta en audio.

4.4. Presentación de la respuesta

- El texto se muestra en pantalla.
- El audio se reproduce automáticamente.

5. Conclusión

Este proyecto permitió integrar múltiples tecnologías para lograr un chatbot con interacción natural en texto y voz, desplegado de manera accesible en la nube.

La implementación demostró la utilidad de **Streamlit** como herramienta para construir aplicaciones web de manera rápida y eficiente, mientras que el uso de **LangChain** junto con el modelo **DeepSeek** permitió dotar al chatbot de una personalidad definida y coherente.

Adicionalmente, la incorporación de **gTTS** y el reconocimiento de voz ofrecieron una experiencia más inmersiva, ampliando las formas de interacción entre el usuario y la aplicación. Gracias a estas integraciones, el chatbot no solo responde preguntas, sino que también transmite una identidad propia inspirada en Javier Santaolalla, lo que genera un mayor impacto educativo y una conexión más cercana con el usuario.

Finalmente, el despliegue en **Streamlit Cloud** evidenció la facilidad de publicar y compartir proyectos en línea, lo cual es esencial en entornos académicos y de investigación. Este tipo de desarrollos abre la puerta a futuras aplicaciones más complejas en el campo de los sistemas digitales y la divulgación científica, combinando accesibilidad, innovación y personalización.