Création d'un Système de Dialogue avec des Sites Web en Utilisant LangChain et Streamlit

Riad Boute

January 29, 2025

Abstract

Cet article présente un système de dialogue intégré avec des sites web en utilisant les bibliothèques `LangChain` et `Streamlit`. Le système permet à l'utilisateur de poser des questions sur un site web spécifique en générant un contexte pertinent à partir de son contenu. Nous détaillons l'architecture de l'application, y compris la configuration de l'historique des conversations, l'extraction et la vectorisation du contenu des sites web, ainsi que le traitement des requêtes utilisateur pour fournir des réponses contextualisées.

1 Introduction

Les assistants virtuels modernes sont conçus pour interagir de manière contextuelle avec les utilisateurs. En intégrant des modèles de langage et des technologies de récupération d'informations, il est possible de créer des systèmes de dialogue intelligents capables de répondre à des questions basées sur des sources d'informations variées. Ce projet utilise la bibliothèque `LangChain` pour le traitement du langage naturel et `Streamlit` pour l'interface utilisateur. L'objectif est de permettre à l'utilisateur de poser des questions sur un site web spécifique, dont le contenu est analysé et utilisé pour fournir des réponses pertinentes.

2 Architecture du Système

Le système repose sur plusieurs composants clés :

- **Chargement de contenu web** : L'application utilise la classe `WebBaseLoader` pour extraire le contenu d'un site web donné.
- **Diviser le texte en morceaux** : Le texte extrait est ensuite divisé en morceaux plus petits à l'aide de `RecursiveCharacterTextSplitter`, afin de faciliter la gestion de documents volumineux.
- **Vectorisation et stockage** : Le contenu est vectorisé à l'aide d'OpenAI Embeddings et stocké dans une base de données vectorielle `Chroma`.
- **Chaîne de récupération contextuelle** : Une fois le contenu analysé et stocké, une chaîne de récupération est créée pour générer des réponses aux requêtes utilisateur en utilisant un modèle de langage comme `ChatOpenAI`.
- **Interface utilisateur Streamlit** : `Streamlit` est utilisé pour créer l'interface interactive où les utilisateurs peuvent entrer une URL de site web et poser des questions.

3 Fonctionnalités du Code

Le code est structuré de manière modulaire, permettant de gérer chaque étape du processus de manière indépendante et claire. Les principales fonctionnalités sont décrites ci-dessous :

3.1 Chargement du contenu d'un site web

La fonction `get_vector_store_from_url` est responsable de charger le contenu d'un site web donné. Elle utilise la classe `WebBaseLoader` pour extraire les documents et les découper en morceaux de texte à l'aide de `RecursiveCharacterTextSplitter`. Le texte est ensuite vectorisé avec `OpenAIEmbeddings` et stocké dans une base de données `Chroma`.

3.2 Création de la chaîne de récupération contextuelle

La fonction `get_context_retriever_chain` crée une chaîne de récupération de contexte en utilisant un modèle de langage (`ChatOpenAI`). Elle génère une requête de recherche en fonction de l'historique de la conversation et du texte contextuel extrait des documents. Cette chaîne permet de récupérer des informations pertinentes pour répondre à une question posée par l'utilisateur.

3.3 Réponse aux requêtes utilisateur

Lorsqu'un utilisateur pose une question, la fonction `get_response` est appelée pour générer une réponse. Elle utilise la chaîne de récupération contextuelle pour obtenir le contexte nécessaire et passe la requête à un modèle de langage afin de produire une réponse adaptée à la question.

3.4 Interface utilisateur Streamlit

L'interface utilisateur permet à l'utilisateur de saisir une URL de site web et de poser des questions. Le contenu du site est d'abord chargé et analysé, puis les questions sont traitées en temps réel. L'historique des conversations est maintenu à l'aide de la fonctionnalité `st.session_state`, permettant une interaction fluide et naturelle.

4 Explication détaillée du Code

4.1 Chargement des données depuis un site web

Le code commence par charger les bibliothèques nécessaires, notamment `langchain`, `streamlit`, et `dotenv`. La fonction `get_vector_store_from_url` est appelée pour récupérer le contenu d'un site web. Si le site est correctement chargé, il est découpé en morceaux et stocké dans une base de données vectorielle. La fonction retourne ensuite la base de données pour être utilisée dans la chaîne de récupération.

4.2 Création d'une chaîne de récupération et de génération de réponses

La fonction `get_conversational_rag_chain` construit une chaîne qui intègre la récupération de contexte et la génération de réponses. Elle crée une chaîne de récupération historique, en intégrant l'historique de la conversation dans le prompt. Une fois que le contexte est extrait, il est intégré à la chaîne de génération de réponses à l'aide d'un modèle `ChatOpenAI`.

4.3 Interaction avec l'utilisateur

L'interface utilisateur permet de saisir une URL et de poser des questions. Lorsqu'une question est posée, l'historique de la conversation est mis à jour, et une réponse est générée en temps réel en utilisant la chaîne de récupération et de génération. Les réponses sont ensuite affichées à l'utilisateur.

5 Conclusion

Ce système offre une manière simple et efficace d'interagir avec n'importe quel site web à l'aide de modèles de langage et de bases de données vectorielles. En combinant `LangChain` et `Streamlit`, il est possible de créer des assistants virtuels capables de répondre à des questions basées sur des contenus web, ce qui ouvre la voie à de nombreuses applications intéressantes dans divers domaines, comme l'éducation, la recherche, et le support client.