8INF974 – ATELIER PRATIQUE EN INTELLIGENCE ARTIFICIELLE II

Projet #1 – Élaboration d'un Chatbot avec Langchain

Objectifs du projet :

- Prendre en main l'environnement de travail à l'UQAC
- Explorer la mise en œuvre de chatbots avec LangChain
- Intégrer l'intelligence artificielle dans un vrai logiciel
- Comprendre la relation entre la théorie vue en classe et l'implantation réelle
- Approfondir ses connaissances de manière autodidacte
- Acquérir de l'expertise dans le développement d'intelligences artificielles

Mise en situation :

L'Université du Québec à Chicoutimi (UQAC) a été fondée en 1969 et elle regroupe plus de 6500 étudiants annuellement répartis au niveau de plus de 200 programmes d'études différents. Afin de maintenir une cohésion dans l'administration de l'organisation, l'Université a créé au fil du temps un manuel de gestion qui regroupent tous les règlements, politiques et procédures afin d'assurer la seine gestion de l'organisation.

Or, le manuel est devenu excessivement volumineux avec le temps et il est de plus en plus difficile de retrouver l'information rapidement pour les employées de l'université. À ce moment-là, il est de plus en plus difficile de fournir une information claire et efficace aux étudiants. À la suite des récents développements liés au traitement du langage naturel (NLP) et à l'apprentissage automatique (ML), il a été mandaté aux étudiants du cours de faire une preuve de concept pour l'élaboration d'un chatbot qui permettrait de répondre efficacement aux interrogations des employés en ce qui a trait au manuel de gestion.

Travail à effectuer :

À l'aide de la librairie Python Langchain, vous devez faire l'implémentation d'un chatbot en utilisant la technique RAG (*"Retrieval Augmented Generation"*). L'information technique pour exécuter cette tâche se trouve sur le site Web de Langchain. La bibliothèque de l'UQAC vous fournit gratuitement le livre Generative AI with LangChain publié à la fin de 2023 (Attention ! les exemples de codes ne sont plus à jour). Pour des raisons de confidentialités, de protection des données et surtout de coûts, vous devez utiliser **un modèle téléchargé localement** pour réaliser votre projet. Vous pouvez trouver un modèle sur le site internet GPT4All ou sur une autre plateforme similaire. Vous être libre de choisir votre modèle (recommandation : Mistral Small).

Au niveau des documents qui sont inclus dans le manuel de gestion, vous pouvez trouver l'information sur le site internet de l'UQAC avec l'url suivant : https://www.uqac.ca/mgestion/. Le manuel de gestion est très volumineux et comporte plus de 200 liens urls qui mènent à des pages HTML ou des documents PDF uniques. Vous devez implémenter dans votre code une technique de moissonnage (web scrapping) qui va aller chercher tous les liens pertinents et de mettre en mémoire l'information pertinente. Au niveau des pages html, le contenu pertinent à utiliser pour votre RAG sont mis dans des HTML tags ayant comme nom de classe "entry-header" et "entry-content". Pour les fichiers PDF, vous devez downloader temporairement le fichier localement (librairie tempfile) et faire l'extraction des données. Ensuite, vous devez persister localement les données à l'intérieur d'une base de données. Vous sauverez ainsi beaucoup de temps, car l'importation est coûteuse en temps en raison du volume. Langchain fourni plusieurs possibilités pour implémenter cette DB et la persistance est très simple à faire.

Au niveau du chatbot, vous devez le créer de tel sorte que l'utilisateur demande une question à propos du guide de gestion et le bot fournira la réponse à l'écran. La réponse **doit donner la source** d'où provient l'information qu'il a donné sous la forme d'un lien URL. Par exemple, si la réponse provient de la « *Politique relative à la planification stratégique* »

le chatbot doit mentionner : "source : https://www.uqac.ca/mgestion/chapitre-2/reglement-sur-la-mission-et-les-valeurs-de-luqac/politique-relative-a-la-planification-institutionnelle/". Enfin, vous devez implémenter un mécanisme de mémoire pour que le bot utilise les questions et les réponses précédentes pour répondre à de nouvelles questions.

Pour le frontend de l'application, commencez simplement par utiliser la console ligne de commande de Python. Par la suite, si votre projet avance bien, utilisé un framework web qui va gérer pour vous le CSS et le Javascript en arrière-plan. Streamlit est une bonne option si vous voulez l'implémenter, car LangChain a déjà développé des agents et des objets pouvant intégrer facilement votre RAG à l'intérieur de ce framework (e.g., l'objet SteamlitChatMessageHistory).

Pour ce projet, on s'attend à l'équivalent de **30 heures de travail par personne** sur 5 semaines (2 et demi en semestre condensé d'été).

Rencontres avec le professeur :

Vous devez être préparés pour vos rencontres avec le professeur. Celles-ci devraient se dérouler de cette façon :

1. Récapitulatif d'où vous en étiez précédemment
2. Présentation de vos travaux respectifs
3. Présentation de votre plan et des problèmes auxquels vous faites face
4. Apprentissage fait depuis la dernière rencontre (le cas échéant)

Rapport de projet :

Pour vos rapports de projet, vous devriez être en mesure de succinctement :

1. Présenter les points clés de votre implémentation
2. Parler de ce que vous avez réussi et ce qui a échoué
3. Faire un bilan des bugs/défis/apprentissages (e.g., ce que vous feriez différemment)
4. Montrer des exemples et des captures d'écran
5. Montrer des bouts de code si pertinent