

Rapport sur le projet Footinator

Yann EDIKEU – Olivier MANYIM

Le projet Footinator vise à créer un chatbot interactif capable de deviner le joueur de football en activité auquel l'utilisateur pense en posant une série de questions. Il nous a été inspiré en grande partie par Akinator dans la logique.

Partie interface graphique :

Aperçu

Nous avons essayé d'implémenter la partie interface graphique. Cependant, nous n'avons pas pu la compléter. Dans la partie interface graphique, une page d'accueil s'affiche avec un bouton "commencer". Il suffit de cliquer sur le bouton commencer, et une question aléatoire apparaîtra pour déterminer l'identité du joueur. Selon la question posée, il suffit de cliquer sur les boutons "oui" ou "non". Cependant, les boutons oui et non ne marchent car nous n'avons pas pu faire le filtrage côté graphique.

comment lancer la page du jeu?

Pour lancer la page du jeu, il faut d'abord se connecter à postgresql. Pour cela vous devez d'abord entrer dans le répertoire footballator. Puis lancer cette commande : `sudo -u postgres psql`. Une fois la commande lancée, veuillez entrer votre mot de passe de votre système. Maintenant que vous êtes dans postgres, il faut charger le fichier init.sql en lançant cette commande : `\i init.sql`. Cela créera et remplira automatiquement les tables nécessaires. joueurs et questions. Une fois que cela est fait vous pouvez vérifier que les tables sont bien disponibles en faisant la commande : `\dt`. Après cela vous pouvez vous déconnecter de postgresql en faisant : `ctrl+d`. et dans le terminal, lancer le serveur en tapant : `node server.js`. Le serveur sera lancé, puis allez dans votre navigateur et tapé : `localhost:3000` et vous accéderez à la page du chatBot

Partie terminale :

Pour cette partie nous avons choisi d'implémenter notre chatbot en utilisant le langage de programmation Python. La base de connaissances(le fichier `knowledge_base`) du chatbot a été stockée dans un fichier JSON contenant une liste de questions, de réponses et de formats de réponse associés. Nous avons utilisé

des techniques de traitement du langage naturel (NLP) pour trouver la question la plus proche de la réponse de l'utilisateur et pour filtrer les joueurs en fonction des critères fournis. Nous avons également utilisé des algorithmes de recherche de mots similaires pour améliorer la convivialité du chatbot. Enfin nous utilisons un fichier csv qui a été traité au préalable afin d'y effectuer les filtres et trouver le joueur correspondant au final.

Il est également possible d'ajouter des réponses que la base de connaissance ne saurait traiter, grâce à toutes les réponses déjà enregistrées précédemment.

Du point de vue quantitatif, le chatbot a montré une capacité raisonnable à filtrer les joueurs en fonction des critères spécifiques. Les réponses fournies par le chatbot étaient en accord avec les critères que nous lui fournissons ce qui indique une précision acceptable.

Du point de vue qualitatif, le chatbot est appréciable pour sa convivialité et sa capacité à poser des questions pertinentes. On peut noter que le chatbot a une compréhension adéquate des réponses fournies, même dans les cas où les réponses n'étaient pas directement liées aux formats de réponse existants. A noter que dès lors que l'une des informations entrées par l'utilisateur est fausse, tout le reste de la recherche se retrouve fausser malheureusement.

- Une amélioration possible consisterait à demander à un utilisateur qui pense à un joueur qui n'est pas contenu dans notre fichier csv, des informations supplémentaires en plus de celles qu'il a déjà entrées.

Pour lancer la version terminal -> { cd mychatbot/; python3 main.py; }