Projet réalisé par Killian CHAMBELLANT, Nicolas DOUCHIN et Anaëlle MORIN à la demande de M. Nicolas DELESTRE dans le cadre des projets SOSI.

But du projet

Le but de ce projet est de créer un mini assistant capable de répondre à une question qui lui a été posée en langue naturelle. Dans un premier temps, cette question doit être reformulée par le programme pour devenir une requête formelle permettant d'interrogée la base de données. La réponse fournie doit être récupérée par le programme dans un type de base de données particulière du web appelée le web des données liées.

Ce genre de programme a déjà été traité par le passé. Il nous a donc été demandé d'utiliser le framework Python Quepy qui permet déjà de transformer une question posée en langage naturelle en une requête formelle. À partir de cela trois prototypes étaient à fournir pour explorer différentes possibilités :

— La première version devait reprendre le travail réalisé par Machinalis et fonctionné comme le site `http://quepy.machinalis.com/`. On avait ici une question en anglais qui était traitée et la réponse cherchée dans la base DBPedia.
— La deuxième version devait permettre de poser les questions en français au lieu de l'anglais.
— Enfin la troisième version devait requêter la base de données Wikidata.

Il est à noter que le développement de Quepy a été arrêté depuis deux ans.

Protoype 1

Protoype 2

Le but de ce prototype est de reprendre le premier afin de pouvoir poser des questions en français.

Pour pouvoir traiter une question en français il faut modifier le tokenizer qui gère le passage des questions en anglais (langage naturel) vers un langage formel. Quepy utilise nativement nltkdata. Nous avons regardé s'il était possible d'utiliser un autre tokenizer qui aurait pu être soit une version française de nltk soit un autre. Malheureusement, après analyse du code de quepy, nous en sommes venus à la conclusion que quepy était intrinsèquement lié à nltk (il y a un module python qui s'occupe de la liaison avec nltk qui est directement inclus dans quepy). L'objectif du projet ne prenait pas en compte la modification de quepy. Et concernant nltk français, pour configurer quepy il aurait également fallu modifier quepy. Pour réaliser le prototype 2, nous avons décidé d'utiliser une solution tierce non viable sur le long terme qui est de traduire avec google traduction les questions posées par l'utilisateur puis d'envoyer le résultat à quepy. Comme dit précédemment, cette solution n'est pas viable sur le long terme car nous pourrions avoir des erreurs de traductions sur des questions plus complexes. À noter également que google traduction gère correctement le passage du français vers l'anglais, mais que cela n'est pas vrai pour toutes les langues. À terme, il sera donc nécessaire d'examiner une solution via un tokenizer pour gérer le support linguistique du programme.

Prototype 3

Le principe de ce prototype est de reprendre le programme et de l'adapter afin d'interroger la base de données "Wikidata".

Wikidata est une base de données linguistiquement neutre et est basée sur des faits et non pas sur des opinions comme DBPedia qui est alimentée par Wikipédia.

Le choix d'implémentation de base de quepy fait qu'il renvoie une chaine de caractères et non pas l'URI. Alors que nous voulions récupérer l'URI afin de pouvoir réaliser des requêtes sur des éléments précis et de pouvoir moduler les réponses. Pour pouvoir récupérer l'URI nous avons été obligés de modifier la requête générée à la volée avant qu'elle soit envoyée à Wikidata. Grâce à l'URI récupérée et aux metadatas nous générons d'autres requêtes pour obtenir les informations pertinentes recherchées. Pour cela, nous avons ajouté un package contenant des modules permettant facilement l'ajout de types de question. Notamment par la déclaration de méthodes selon la metadata obtenue.

Pistes d'amélioration

Notre solution, bien que fonctionnelle pour quelques questions, présente de différents défauts. Tout d'abord, quelque soit Il aurait fallu que quepy propose un plus grand paramétrage pour la création de la requête et l'utilisation du tokenizer.