Document de conception

Projet d'interopérabilité

Groupe 4

Idriss BENGUEZZOU
Iness BOUABID
Ali BOUGASSAA
Ghilas MEZIANE

01/03/2023

Table des matières

1 Introduction

1.1 Objet du document

Ce document de conception a pour objectif de décrire les spécifications techniques et fonctionnelles de wikistone. Il présente une vue d'ensemble du système et détaille les différentes fonctionnalités qu'il offre, ainsi que les interactions entre elles. Ce document s'adresse aux développeurs, aux concepteurs et à toute personne impliquée dans le développement ou l'utilisation du système. Il fournit une référence complète pour comprendre les choix de conception, les fonctionnalités et les contraintes techniques du projet, ainsi qu'une base pour la mise en œuvre et la maintenance future.

1.2 Contexte et objectifs du projet

Le projet Wikistone a été initié pour répondre au besoin de centraliser et de faciliter l'accès à des données sur les monuments historiques de la région de la Loire. Les données sur ces monuments sont souvent éparpillées sur de nombreux sites et sources, ce qui rend leur accès et leur utilisation difficile. L'objectif principal de Wikistone est de mettre en place une base de connaissances interopérable pour stocker et maintenir à jour les données sur les monuments historiques, en utilisant Wikibase comme système de gestion de données.

Pour atteindre cet objectif, deux applications principales ont été développées : Wiki Injector pour l'ingestion de données, et Wiki Interface pour la recherche et la consultation de données. L'application Wiki Injector permet d'extraire, d'analyser et d'injecter des données dans Wikibase, tandis que Wiki Interface permet de rechercher et de consulter les données stockées dans la base de connaissances.

Le projet Wikistone a été développé en utilisant principalement le langage de programmation Python et Javascript, avec l'utilisation de différentes bibliothèques et outils pour l'analyse et la manipulation de données.

Dans la suite de ce document, nous détaillerons les différentes parties du projet Wikistone, en commençant par une présentation de Wikibase et de son utilisation dans le projet, suivie d'une description des applications de wikistone.

2 Wikistone : une base de connaissances interopérable

2.1 Présentation de Wikibase et de son utilisation dans le projet

Wikibase est un logiciel libre permettant la création de bases de connaissances structurées et collaboratives. Il a été développé par la Wikimedia Foundation pour gérer Wikidata, la base de données centrale de connaissances de l'écosystème Wikimedia.

Dans le projet Wikistone, Wikibase est utilisé pour créer une base de connaissances interopérable, qui permettra de relier des données de sources différentes.

Pour permettre de déployer Wikibase et ses dépendances dans un environnement Docker, de manière à faciliter son installation et sa configuration, un fichier docker-compose.yml est présent dans notre projet et définit les services nécessaires au bon fonctionnement de Wikistone.

Plus précisément, deux applications de notre projet interagissent avec Wikibase de différentes manières.

D'une part, Wiki Interface utilise l'endpoint :
http://localhost:8834/proxy/wdqs/bigdata/namespace/wdq/sparql pour réaliser des requêtes SPARQL sur notre base de connaissances.
D'autre part, Wiki Injector utilise l'API http://localhost/w/api.php pour injecter des données dans la base de connaissances.

Ces deux méthodes d'interaction avec Wikibase permettent ainsi à notre projet de manipuler efficacement les données et de les rendre facilement accessibles pour les utilisateurs.

3 Les applications de Wikistone

3.1 Présentation de Wiki Injector et de Wiki Interface

Wiki Injector est l'application chargée d'extraire, d'analyser et d'injecter des données dans Wikibase. Elle est principalement utilisée pour collecter des données relatives aux monuments historiques à partir de différentes sources, telles que des fichiers CSV, des APIs ou des sites Web en ligne. Elle est également responsable de la préparation des données pour qu'elles soient compatibles avec la structure de Wikibase.

Wiki Interface, quant à elle, est l'application qui permet la recherche et la consultation de données dans Wikibase. Elle offre une interface utilisateur conviviale qui permet de rechercher des monuments historiques selon différents critères, tels que leurs localisations géographiques, leurs catégories, leurs noms, etc. Elle permet également d'afficher les détails de chaque monument.

Dans les sections suivantes, nous allons détailler l'architecture, les fonctionnalités et les technologies utilisées pour ces deux applications.

4 Wiki Injector : une application pour l'ingestion de données

4.1 Architecture de Wiki Injector

L'architecture de Wiki Injector est divisée en trois parties principales, chacune implémentant une fonction spécifique.

1. **Le module Data Formatter :** Ce module permet de formater les données provenant de différentes sources externes en un objet JSON commun en Python.

2. **Le module Classifier :** Ce module effectue la classification des données en utilisant un algorithme de machine learning. Il est composé de deux sous-modules :

   — **Le sous-module Clustering Data :** qui implémente différentes techniques de clustering (telles que KMeans, Agglomerative Clustering, DBScan, etc.) pour regrouper les données brutes en clusters.

— **Le sous-module Classify Data :** qui implémente différents algorithmes de machine learning (tels que Decision tree, Gaussian SVM, KNN, Multinomial NB...) pour prédire la classe des monuments à partir des données d'entrainement issue du clustering. Le meilleur modèle est ensuite entraîné et utilisé pour prédire les classes de monuments.

3. **Le module Data Injector :** Ce module permet l'ajout de nouvelles données dans la base de données Wikibase. Il prend en entrée les fichiers JSON formatés par le module Data Formatter, et utilise l'API Python Wikibase pour se connecter à Wikibase et y insérer les données.

Jusqu'ici Wiki injector était une application en ligne de commande destinée principalement à des personnes familières à ces environnements. C'est pourquoi nous avons décidé de mettre en place également Wiki Admin.

Wiki Admin est une application développée en ElectronJS qui permet de faciliter l'ingestion de données dans notre base de données Wikibase Elle a été conçue pour être utilisée par les administrateurs de la base de données afin de mettre à jour et d'injecter de nouvelles données.

L'application est principalement développée en HTML, CSS et JavaScript, et elle utilise Eel pour communiquer avec les fonctions définies dans notre programme Python. Elle dispose également d'une interface utilisateur intuitive, qui permet à l'utilisateur de gérer les propriétés présentes dans la Wikibase.

4.2 Extraction de données

Le processus d'extraction de données de l'application Wiki Injector s'appuie sur trois sources principales : le site web https://www.loire.fr/, la récupération d'un fichier CSV à partir de https://dataclic.fr/ et l'API https://data.culture.gouv.fr/api/v2/console.

Les données extraites à partir de ces sources sont collectées par les différents programmes présents dans le module data_formatter. Le programme convertit en un format JSON, les données brutes extraites pour permettre le traitement ultérieur des données.

Pour avoir tous les détails concernant le processus d'extraction et de convertion des données, se référer au document "**Extraction d'information et ingestion**", section *Principe d'extraction d'informations.*

De plus, pour comprendre la structure des données extraites voir le document "**Définition et description des sources d'information**".

4.3 Analyse de données

La problématique qui se posait était que les monuments présents dans les données provenant de nos sources ne possédaient pas l'information concernant la catégorie du monument, c'est pourquoi nous avons décidé de mettre en place un moyen permettant d'automatiser la classification des monuments.

Nous avons décidé d'utiliser une approche utilisant le machine learning. La première étape a été de concevoir un outil permettant de déterminer les différentes classes présentes dans un jeu de données. C'est alors que nous avons mis en place le module **clustering_data**, sous la forme d'un notebook.

Nous avons utilisé un grand dataset de plus de 5000 monuments pour clustériser et extraire des données labelisées, afin d'obtenir des résultats plus précis sur le classifier que nous allions entraîner.

Ne connaissant pas les catégories au préalable des monuments présents dans notre jeu de données, ni le nombre de catégories présentes, il a fallu utiliser différentes méthodes pour les déterminer. Nous avons opté pour la méthode du coude (Elbow Method) pour déterminer le nombre optimal de clusters à utiliser dans l'algorithme de clustering.

Enfin, après avoir testé plusieurs matrices de similarité (entre autre Word2Vec), nous avons observé que les meilleurs résultats étaient obtenus avec la matrice TFIDF et l'algorithme KMeans.

Après avoir choisi la matrice TFIDF et l'algorithme KMeans pour notre clustering, nous avons poursuivi notre analyse en testant plusieurs classifiers à savoir le Decision Tree, le Gaussian SVM, le K-nearest neighbors, et le Multinomial NB. Après avoir entraîné et testé chaque modèle, nous avons remarqué que tous les modèles étaient très performants sur nos données de tests, avec une légère supériorité pour le Decision Tree qui a obtenu le meilleur score.

Nous avons ensuite entraîné notre modèle sur les monuments clusterisés et labellisés pour prédire les classes des monuments des données que nous utilisons pour alimenter notre wikibase. Les données d'entraînement utilisées pour notre modèle de classification de monuments se trouvent dans le fichier "donnees_clusterisees.json".

Le modèle de classification maintenant fonctionnel peut être utilisé depuis notre application d'injection pour labeliser nos données une fois extraites.

4.4 Injection de données dans Wikibase

Le processus d'ingestion des données dans Wikibase se déroule en deux étapes. La première étape consiste à vérifier si les propriétés existent déjà dans Wikibase et, si nécessaire, à les créer ou à les mettre à jour avec les nouvelles étiquettes et descriptions.

La deuxième étape consiste à parcourir chaque élément du jeu de données et à vérifier s'il existe déjà dans Wikibase. Si l'élément existe, la fonction met à jour ses propriétés et déclarations. Si l'élément n'existe pas, la fonction crée un nouvel élément avec les propriétés et valeurs spécifiées.

Pour plus de détails sur ce processus, veuillez consulter le fichier "**Extraction d'information et ingestion**" section "*Principe d'ingestion des informations*".

5 Wiki Admin : une application d'administration

Wiki Admin est à la fois une application web et de bureau. Cela offre une grande flexibilité aux utilisateurs pour gérer et administrer la Wikibase en fonction de leurs besoins et de leurs préférences d'utilisation. Elle est développée à l'aide des technologies ElectronJS, HTML, CSS et JavaScript.

Wiki Admin permet d'injecter des données dans la Wikibase à l'aide des modules **wiki_injector** et **classifier**, et de les mettre à jour si nécessaire. L'application permet également de gérer les propriétés présentes dans la Wikibase, notamment de les modifier, de les supprimer ou d'en créer de nouvelles.

Les modules **wiki_injector** et **classifier** sont intégrés à l'application grâce à la bibliothèque Python Eel (*Embedded Browser Python Library*).

Eel permet de créer des applications de bureau et web en utilisant Python pour le backend et des technologies web (HTML, CSS, JS) pour le frontend. En utilisant Eel, les modules Python peuvent être appelés directement depuis l'interface web de l'application. Dans le cas de Wiki Admin, cela signifie que les fonctionnalités de **wiki_injector** et **classifier** peuvent être utilisées directement depuis l'interface de l'application, sans avoir besoin d'écrire du code Python ou de lancer des scripts en dehors de l'application.

Cela rend l'utilisation des modules plus accessible aux utilisateurs qui ne sont pas familiers avec la programmation, et permet d'intégrer facilement des fonctionnalités avancées à l'interface de l'application.

Enfin, Wiki Admin est conçue pour être facilement utilisable, même par des utilisateurs qui ne sont pas familiers avec la Wikibase ou les technologies utilisées pour son administration. Elle possède une interface graphique intuitive, qui permet de naviguer facilement entre les différentes sections et de réaliser les opérations souhaitées en quelques clics.

6 Wiki Interface : une application pour la recherche et la consultation de données

6.1 Architecture de Wiki Interface

Wiki Interface est une application web développée à l'aide de Python et du framework Flask. Elle utilise également Jinja pour le templating, du CSS pour le style et du JS pour la réalisation de requêtes SPARQL. Pour l'affichage de la carte du monde, l'application utilise la bibliothèque Leaflet.js et pour la transformation d'adresse en coordonnées géographiques, elle utilise l'API MapQuest.

L'application est divisée en plusieurs couches : la couche de présentation et la couche de logique métier. La couche de présentation est chargée d'afficher les résultats de la recherche sous forme de cartes. La couche de logique métier est responsable de l'exécution des requêtes SPARQL et du traitement des résultats.

6.2 Fonctionnalités de recherche et de consultation

Wiki Interface offre plusieurs fonctionnalités de recherche et de consultation des données stockées dans la Wikibase. Les utilisateurs peuvent effectuer des recherches textuelles sur les entités stockées dans la base de données, et filtrer les résultats en fonction de différents critères, tels que la localisation, le code postal, la region, la catégorie du monument... L'application permet également d'afficher les résultats de la recherche sur une carte interactive, où chaque entité est représentée par un marqueur. Les utilisateurs peuvent également visualiser les informations de chaque entité dans une page dédiée.

7 Conclusion

7.1 Synthèse du document de conception

La conception du projet Wikistone a permis de créer deux applications fonctionnelles et complémentaires : Wiki Admin, pour la gestion et l'administration de la base de données Wikibase, et Wiki Interface, pour la recherche et la consultation des données stockées dans cette même base. Cette approche complète permet de répondre aux différents besoins des utilisateurs en matière de gestion, de recherche et de consultation de données liées aux monuments historiques.

Avec ses fonctionnalités de gestion de propriétés, d'injection et de mise à jour de données, ainsi que ses fonctionnalités de recherche textuelle, de filtrage des résultats et de visualisation sur une carte interactive, Wikistone offre une solution complète et intuitive pour l'organisation et la gestion des données. Cette solution répond aux attentes des utilisateurs et leur permet d'explorer efficacement et facilement les données relatives aux monuments historiques.

En somme, Wikistone est un projet complet et fonctionnel qui répond aux besoins des utilisateurs en matière de gestion et de consultation de données liées aux monuments historiques. La prochaine étape consistera à améliorer et à développer ces applications en fonction des besoins des utilisateurs.

7.2 Perspectives d'avenir pour le projet Wikistone

Le projet Wikistone a un grand potentiel d'avenir, car il répond à un besoin croissant de gestion et de conservation des données patrimoniales. Les perspectives d'avenir pour le projet incluent :

— L'élargissement de la base de données en ajoutant de nouveaux types de monuments ou de données patrimoniales, tels que des informations sur les sites archéologiques, les œuvres d'art, etc.

— L'implémentation d'un système de recommandation pour aider les utilisateurs à découvrir de nouveaux monuments historiques ou patrimoniaux en fonction de leurs préférences et de leur historique de recherche.

— La mise en place d'un système de traduction automatique pour faciliter l'accès aux informations pour les utilisateurs qui ne parlent pas la langue dans laquelle les données sont initialement entrées.

— La mise en place d'un système de vérification des données pour garantir la qualité et la fiabilité des informations stockées dans la base de données.

— L'ajout de fonctionnalités de contribution collaborative pour permettre aux utilisateurs de participer à la collecte et à la gestion des données, ainsi qu'à la documentation des monuments historiques et patrimoniaux.

— La création d'une version mobile de l'application.

En somme, le projet Wikistone offre une solution innovante et prometteuse pour la gestion de données patrimoniales. Le potentiel de développement est important, et il est possible d'envisager de nombreuses perspectives d'avenir pour cette plateforme collaborative.