Manuel méthodologique du projet web

projet personnel

Xinyi Shen

Master TAL M2

INALCO

Sommaire

Objectif

Ce manuel est une instruction sur l’application Hôtel & Dictionnaire des langues peu dotées qui permet à nos clients d'explorer les possibilités actuellement offertes dans ces deux domaines et de prendre la décision de les investir.

Concrètement, d’une part, l’application vous montrera la possibilité d’un système d’hôtels écologiques . Et d’autre part, l’application vous montrera aussi la possibilité de soutenir la communauté des bénévoles travaillant sur les langues peu dotées, dans l’objectif de créer un marché de la donnée autour de ces langues.

Installation

Pré-requis : outils et installation

- L'environnement virtuel : pipenv
- Les outils pour scrapping : Beautiful Soup, sélénium
- L'outil de l'app : streamlit
- L'outil graphique : matplotlib.pyplot
- L'outil d'unicode : unidecode

Étape 1 : Créer un environnement virtuel python

```
pip3 install pipenv
. venu/bin/activate
```

Étape 2 : Installer tous les outils requis

```
pip3 install -r requirements.txt
```

Étape 3 : Lancer l'application

```
streamlit run app.py
```

Méthodologies scrapy

1. Beautiful Soup

Beautiful Soup est une bibliothèque qui permet de récupérer facilement des informations à partir de pages Web. Il se trouve au sommet d'un analyseur HTML ou XML, fournissant des idiomes pythoniques pour itérer, rechercher et modifier l'arborescence d'analyse.

1.1 Accès le site

```
def getHotelName(country):
    countryName = country.lower()
    countryName = unidecode(countryName)
    url = "https://www.nh-hotels.fr/hotels/" + countryName
    header = {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) App
    response = requests.get(url, headers = header)
    soup = BeautifulSoup(response.content, 'html.parser')
```

Pour accéder le site avec BeautifulSoup, il faut avoir:

- Un url
- un request
- La méthode BeautifulSoup
- headers (ça dépend le siteweb)

1.2 Méthodes d'obtenir des informations

```
    #scraping the info

    #hotel name
    title = soup.find('title').contents[0].split("|")[0]
    hotel= title
```

```
    if color_primary.select_one('img[alt="Eco-friendly"]'):
        ecofriendly = "Oui"
    else:
        ecofriendly =  "Non"
```

```
for eachHotel in soup.find_all("div", class_="block-body"):
    for all_a in eachHotel.find_all("a"):
        hotelUrl = all_a.get("href")
        hotels.append(hotelUrl)
```

Pour obtenir les informations dont nous avons besoin, BeautifulSoup offre plusieurs méthodes. Ici, ce que j'ai utilisé c'était de:

- find : trouver l'élément
- select_one : sélectionner un objet particulier
- get("href") : obtenir les urls
- find_all : trouver tous les éléments qui ont le même nom

2. Selenium

Les liaisons Selenium Python fournissent une API simple pour écrire des tests fonctionnels / d'acceptation à l'aide de Selenium WebDriver. Grâce à l'API Selenium Python, vous pouvez accéder à toutes les fonctionnalités de Selenium WebDriver de manière intuitive.

Les liaisons Selenium Python fournissent une API pratique pour accéder à Selenium WebDrivers comme Firefox, Ie, Chrome, Remote, etc. Les versions actuelles de Python prises en charge sont 3.5 et supérieures.

1.1 Accès le site

```
option = webdriver.ChromeOptions()
option.add_argument("headless")
driver = webdriver.Chrome(chrome_options=option)
driver.get("https://ntealan.net")
```

Afin d'accéder au site avec selenium, il faut créer un webdriver et puis utiliser la méthode "get" pour ouvrir le site.

La méthode "option" ici, c'est pour ne pas ouvrir le navigateur.

1.2 **Méthodes d'obtenir des informations**

```
try:
    windows = driver.find_element_by_class_name("modal-bottom")
    windows_button = windows.find_element_by_tag_name("button")
    windows_button.click()
except:
    pass
```

```
for key, value in id_mots.items():
    if word == value:
        item = listeUL.find_element_by_id(key)
        item.click()
```

Pour trouver des informations, il existe plusieurs méthodes:

- find_element_by_class_name

- find_element_by_tag_name
- find_element_by_id
- find_element_by_xpath

Ces méthodes permettent de trouver les éléments par son class, son tag, son id ou son xpath. Pour obtenir le texte, il suffit d'ajouter ".text" à la fin.

La méthode "click()" permet de cliquer sur cet élément et ouvrir une autre page.

Création de l’app

1. Streamlit

Le framework d'applications open source de Streamlit est le moyen le plus simple pour les scientifiques des données et les ingénieurs en apprentissage automatique de créer de belles applications performantes en quelques heures seulement! Le tout en pur Python.

```
st.sidebar.header("À propo")
st.sidebar.write("Cette application vous propose deux options.")
st.sidebar.write("Le premier, les hôtels Eco-friendly dans l'Hotel Group.")
st.sidebar.write("Le deuxième, un dictionaire peu-doté.")
st.sidebar.write("")
status = st.sidebar.radio("Quel page voulez-vous voir ?", ("Hôtels Eco-friendly", "Dic
st.sidebar.write("")
st.sidebar.subheader("Auteur:")
```

```
    st.text("")
    country = st.text_input("Le pays dans lequel vous voulez chercher (en français)")

    if country != '':
        try:
            avec, sans, info = getData()
            st.subheader("Toutes les hôtels dans ce pays:")
            st.table(info)
            st.text("")
            plt.bar(["Hôtels san Eco-friendly", "Hôtels avec Eco-friendly"], [sans, a
            st.subheader("Nous comparons les hôtels qui ont plus de 3 étoiles:")
            st.set_option('deprecation.showPyplotGlobalUse', False)
            st.pyplot()

        except:
            st.error("Désolé(e), ce pays n'a pas d'hôtel dans l'Hotel Group ")
else:
```

```
        word = st.selectbox("Vous voulez chosir quel mot yemba ?",("a", "á", "alá pū"
        #st.write("Vous avez choisi: ", word, "Merci d'être patient(e)")
        if word != '':
            try:
                dic = find_item(word)
                for key, value in dic.items():
                    st.info(
                        '''
                        {} : {}
                        '''.format(key, value)
                        )

            except:
                st.error("Une erreur de connexion. Vous pouvez ressayer.")
```

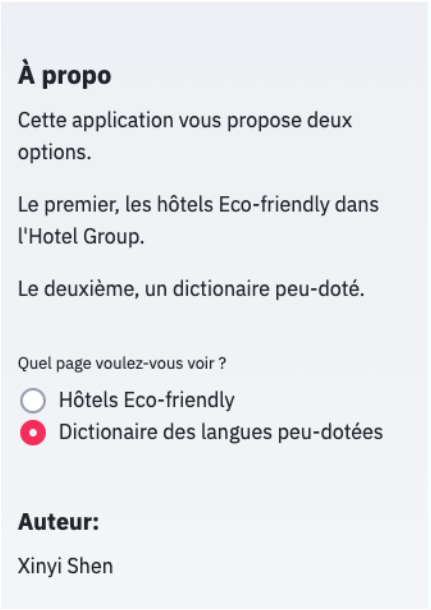
Streamlit est un framework d'application open-source pour les équipes d'apprentissage automatique et de science des données. Il permet de créer l'application en très peu de temps.

Les méthodes basiques:

- st.write() : écrire de l'argument
- st.text() : écrire du texte de largeur fixe et préformaté.
- st.header() : écrire la tête
- st.subheader() : écrire la sub-tête
- st.title() : écrire le titre de l'application
- st.markdown() : afficher la chaîne au format Markdown
- st.pyplot() : afficher la graphe de matplotlib.pyplot
- st.sidebar : chaque élément transmis à st.sidebar est épinglé à gauche, ce qui permet aux utilisateurs de se concentrer sur le contenu de l'application.

2. Visualisation

Après avoir faire tourner le programme, l'application s'ouvre automatiquement : http://localhost:8501



((la visualisation du sidebar. Ici permet de choisir le page)

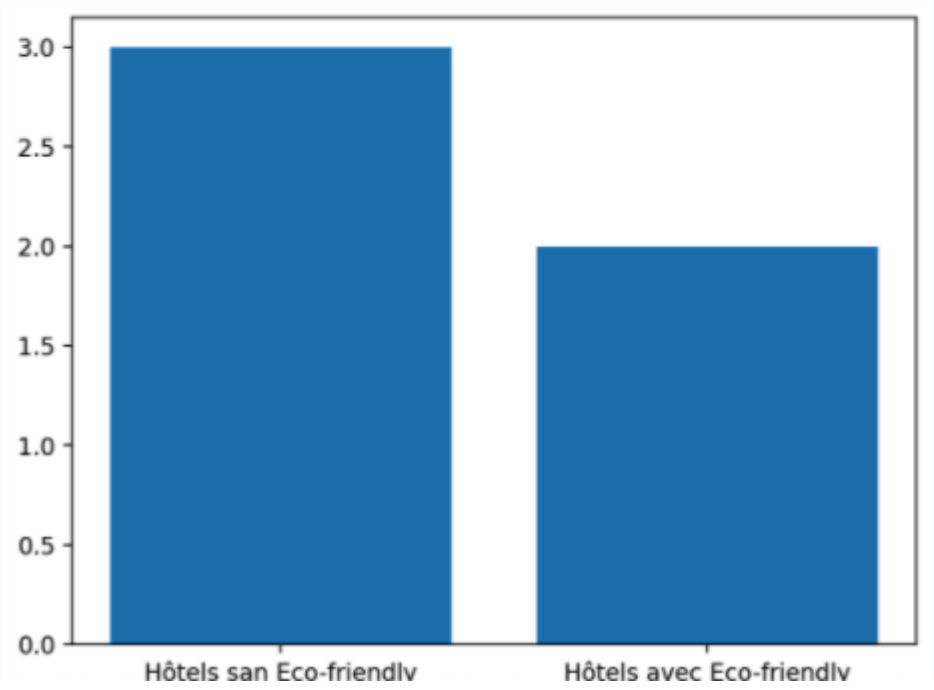
Le pays dans lequel vous voulez chercher (en français)

France

Toutes les hôtels dans ce pays:

	Hôtel	Eco-friendly	Étolies	Plus d'informations
0	Hôtel NH Lyon Airport	Oui	4	https://www.nh-hotels.fr/hotel/nh-lyon-airport
1	Hôtel NH Collection Marseille	Non	4	https://www.nh-hotels.fr/hotel/nh-collection-marseille
2	Hôtel nhow Marseille	Non	4	https://www.nh-hotels.fr/hotel/nhow-marseille
3	Hôtel NH Nice	Oui	4	https://www.nh-hotels.fr/hotel/nh-nice
4	Hôtel NH Toulouse Airport	Non	4	https://www.nh-hotels.fr/hotel/nh-toulouse-airport

Nous comparons les hôtels qui ont plus de 3 étoiles:



(un exemple de la recherche d'hôtel)

(un exemple de l’extraction du dictionnaire des langues peu dotées)