

# Wikipédia : quels enjeux Big Data ?

Benoît MARION

03/03/2024

## Sommaire

<b>1 Mise en contexte</b>	<b>2</b>
1.1 Qu'est-ce que Wikipédia ? . . . . .	2
1.2 Quelques chiffres (au 29 février 2024) . . . . .	2
<b>2 Enjeux Big Data pour Wikipédia</b>	<b>3</b>
2.1 Aspects volumétriques . . . . .	3
2.2 Trafic . . . . .	3
2.3 Cohérence et historicisation . . . . .	3
2.4 Fiabilité du contenu . . . . .	4
2.5 Protection des données personnelles . . . . .	4
2.6 Autres enjeux . . . . .	4
<b>3 Infrastructure de Wikipédia</b>	<b>5</b>
3.1 MediaWiki . . . . .	5
3.2 Bases de données et cache . . . . .	6
3.3 Réseau de distribution de contenu . . . . .	6
3.4 Vérification des données . . . . .	7
3.5 Autres modes d'accès aux données . . . . .	7
3.6 Exemple de parcours de requêtes serveurs depuis le client . . . . .	7
3.6.1 Accès à une page . . . . .	7
3.6.2 Mise à jour d'une page . . . . .	8
3.6.3 Création d'une nouvelle page . . . . .	8
<b>4 Exemple de manipulation de données Big Data sur Wikipédia</b>	<b>8</b>
4.1 Téléchargement des données . . . . .	11
4.2 Récupération de la liste des articles depuis le fichier .xml . . . . .	11
4.2.1 Ressources . . . . .	12
4.2.2 Fonction principale . . . . .	12
4.2.3 Exécution de la fonction . . . . .	12

4.3	Récupération des textes des articles et stockage . . . . .	13
4.3.1	Extraction des textes de articles via <code>wikiextractor</code> depuis le fichier <code>.xml</code>	13
4.3.2	Stockage des fichiers <code>.txt</code> dans une base de données SQL via l'API Python de SQLite . . . . .	15
4.4	<i>GUI</i> simple de recherche et d'affichage des articles de Wikipédia en galicien . . . . .	16
4.4.1	Création de la classe <code>WikiSearch</code> . . . . .	16
4.4.2	Appel de la classe <code>WikiSearch</code> . . . . .	17
4.4.3	Résultat . . . . .	17
4.4.4	Comparaison avec la vraie interface de Wikipédia . . . . .	18
<b>5</b>	<b>Sources</b>	<b>18</b>
5.1	Articles Wikipédia (inception) . . . . .	18
5.2	Sites d'information sur les aspects techniques de wikimédia . . . . .	20
5.3	Téléchargemer les données de Wikipédia . . . . .	20
5.4	Conférences . . . . .	20
5.5	Articles de presse . . . . .	20
5.6	Tutoriels et autres ressources pour extraire les données de Wikipédia . . . . .	20

Ce document a pour objectif de mettre en lumière les enjeux Big Data pour Wikipédia, de présenter l'infrastructure de Wikipédia et de montrer un exemple de manipulation de données Big Data sur Wikipédia.

Les codes utilisés sont disponibles intégralement dans le [dépôt git](#) associé à ce projet.

## 1 Mise en contexte

### 1.1 Qu'est-ce que Wikipédia ?

Wikipédia est une **encyclopédie en ligne, libre, gratuite et collaborative**. Crée le 15 janvier 2001 par Jimmy Wales et Larry Sanger, elle se caractérisa avant tout par son contenu librement réutilisable, son **aspect participatif** (elle est écrite par des bénévoles), sa **structure ouverte** et sa **politique de neutralité** (tant politiquement que pour la publicité).

### 1.2 Quelques chiffres (au 29 février 2024)

Côté contenu accessible, l'encyclopédie Wikipédia est disponible en **326 langues** (contenu différent) et contient plus de **60 millions d'articles** au total. En français, elle contient presque **2,6 millions d'articles** se classant au **quatrième rang** derrière les versions en anglais, cébouano et allemand.

Côté fréquentation, Wikipédia est l'un des sites les plus visités au monde. Le nombre de pages wikipedia francophone visitées serait au delà de **1,3 milliard en janvier 2024**. Avec environ **30 millions de visiteurs uniques** (et plus de 90 millions d'appareils uniques), le site francophone se classe généralement dans le **top 10 des sites les plus visités** en France et dans le top 20 mondial.

Finalement concernant les **modifications**, sur les pages francophones, environ **665 000 pages (soit 1%)** auraient été faites en janvier 2024 et 7 495 pages ajoutées ont été comptabilisées. Pour janvier 2024 , la version francophone est forte d'une communauté de près de **53 000 rédacteurs** (au moins une modification) et de plus de **6 000 contributeurs actifs** (plus de 5 modifications). Concernant, l'ensemble des versions de Wikipédia, on compterait plus de **300 000 contributeurs actifs**, avec 170 000 pages ajoutées, plus de **12 millions de modifications** (en janvier 2024) et environ **2 modifications par seconde en moyenne**.

## 2 Enjeux Big Data pour Wikipédia

### 2.1 Aspects volumétriques

Ces chiffres sont le reflet de la grande quantité de données que se doivent d'héberger les serveurs de Wikipédia. Ainsi le fichier **XML contenant les pages actuelles de Wikipédia en anglais** pèse environ **22,14 Go** (Gigaoctets) sans les médias et compressé. L '**historique complet (avec toutes les discussions, le notes de communautés et les articles)** de **Wikipédia en anglais** pesait environ **10 To** (Téraoctets) en 2013 (pas de données plus récentes trouvées). Le plus gros morceau, **Wikimedia Commons**, qui contient les **images, vidéos et autres médias** utilisés dans les Wikipédias, pèse **428,36 To**.

### 2.2 Trafic

Comme vu précédemment, Wikipédia est un des sites les plus visités au monde. Cela implique un trafic important et une gestion des données en conséquence. En effet, le site doit être capable de gérer des centaines **millions de requêtes** par jour, et ce de manière efficace et rapide. Il se doit aussi de faire face aux **pics de trafic** (comme lors de la mort de personnalités publiques, de catastrophes naturelles, etc.) qui peuvent être très importants et aux **attaques** (comme les attaques par déni de service) qui peuvent être très lourdes.

### 2.3 Cohérence et historicisation

Wikipédia étant avant tout une encyclopédie participative, tout un chacun peut modifier le contenu. Il est donc important de pouvoir garantir la **cohérence des données** entre les différents serveurs et de pouvoir retrouver une **version antérieure** en cas de problème.

## 2.4 Fiabilité du contenu

Wikipédia se doit de garantir la qualité de son contenu. Cela passe par la **vérification des sources**, la **neutralité des articles**, la **suppression des contenus non pertinents ou non vérifiables**, etc. Ce travail est effectué par des bénévoles, mais aussi par des outils automatiques qui peuvent parfois être lourds, notamment dans le cas de “vandalisme” (modification malveillante du contenu) massif ou de “guerre d’édition” (modification répétée d’un contenu) sur de multiples articles dans des contextes politiques tendus (*cf* articles de presse en source).

## 2.5 Protection des données personnelles

Wikipédia se doit de protéger les données personnelles de ses utilisateurs et ce à divers niveau. Cette politique de protection des données personnelles s’explique par différentes raisons :

- **philosophique** : Wikipédia se veut être un projet libre et ouvert, et la protection des données personnelles est un enjeu important pour garantir la liberté des utilisateurs.
- **légale** : Wikipédia se doit de respecter les lois en vigueur sur la protection des données personnelles.
- **technique** : Wikipédia doit protéger les données personnelles de ses utilisateurs contre les attaques informatiques.
- **politique** : Wikipédia doit protéger les données personnelles de ses utilisateurs contre les abus de pouvoir dans certains pays (Wikipédia avait par exemple été interdit d'accès pendant 3 ans en Turquie, *cf* articles de presse en source ).

D'une part, les données personnelles des utilisateurs sont protégées par la **licence Creative Commons** qui interdit l'utilisation des données personnelles à des fins commerciales et aucune donnée personnelle n'est collectée par Wikipédia (sauf pour les utilisateurs enregistrés qui font des modifications).

Aussi, les services de WikiMédia pour éviter les attaques de type *DoS* (déni de service) ou *DDoS* (déni de service distribué) ne peuvent se baser sur les données personnelles des utilisateurs (comme le fait par exemple *Cloudflare*). Enfin Wikipédia offre la possibilité de demander la suppression de données personnelles.

## 2.6 Autres enjeux

D'une part, outre le volume, la **variété des données** est un aspect important. Wikipédia se compose tant de texte structuré que d'images, audios, et vidéos. Beaucoup de ces données sont mutuelles (une image peut être utilisée dans plusieurs articles) et doivent donc être stockées de manière efficace.

D'autre part, Wikipédia étant entièrement gratuite, la **gestion des coûts** est un enjeu important. Il est donc important de pouvoir stocker et traiter les données de manière efficace et peu coûteuse sans gêner la qualité du service pour les utilisateurs.

### 3 Infrastructure de Wikipédia

L'infrastructure de tout Wikipédia est gérée par la **Wikimedia Foundation**, une organisation à but non lucratif possèdant et gérant tous les serveurs sur lesquels sont hébergés les projets Wikimedia. Ils sont installés aux États-Unis, aux Pays-Bas, à Singapour et en France (Marseille). Cette infrastructure relativement complexe est complètement **ouverte et documentée**. Il est possible de proposer des modifications ou des améliorations à l'infrastructure de Wikipédia, et de suivre les évolutions de celle-ci.

L'essentiel des informations sur l'infrastructure de Wikipédia est disponible sur le site **Wikitech** qui est un wiki ouvert à tous. Il est possible d'y trouver des informations sur les serveurs, les logiciels utilisés, les protocoles, les outils, etc. Il est aussi possible de suivre les évolutions de l'infrastructure de Wikipédia via des **tableaux de bord** et des **statistiques** (*cf* sources).

Une **conférence vulgarisant l'infrastructure Wikimedia** tenu en 2019 à la 36C3 (Leipzig) permet de mieux comprendre le fonctionnement et a fortement inspiré cette partie (voir sources).

#### 3.1 MediaWiki

Au cœur de l'infrastructure se trouve **MediaWiki**, le logiciel qui fait tourner Wikipédia. Il s'agit d'un logiciel libre (licence GNU GPL), c'est-à-dire que son code source est ouvert et peut être modifié par n'importe qui pour son utilisation propre ou la publication. Il tourne sur des serveurs **Apache** et est écrit en **PHP** pour afficher les données d'une base de données **MySQL** (ou **MariaDB**). Son rôle est de *parser* les articles en **wikimarkup** pour les afficher en **HTML**. Ces rendus sont automatisés et faits selon les préférences de l'utilisateur (langue, thème, taille de police, etc.) et sont stockés dans les bases de données.

Il permet aussi de faire d'autres actions comme contrôler les utilisateurs, les droits, les pages... Il est aussi capable de gérer des **extensions** pour ajouter des fonctionnalités supplémentaires comme la gestion des médias, des rendus LaTeX ou un algorithme de ML détectant les contributions "malveillantes" ou étant du "vandalisme".

### 3.2 Bases de données et cache

Wikipédia dispose de 6 serveurs au total qui contiennent les données dont voici la liste :

Nom	Localisation	Rôle	Mise en service
Eqiad	Ashburn, Virginia, États-Unis	Application + Cache	2010
Codfw	Carrollton, Texas, États-Unis	Application + Cache	2014
Esams	Amsterdam, Pays-Bas	Cache	2005
Ulsfo	San Francisco, California, États-Unis	Cache	2014
Eqsin	Singapour	Cache	2017
Drmrs	Marseille, France	Cache	2022

On distingue deux types de serveurs : les **serveurs “application”** qui font tourner MediaWiki pour **faire le rendu des pages** et les **serveurs “cache”** qui **contiennent les pages déjà rendues** en HTML dans une base de données. Les premiers comprennent aussi des serveurs de cache. *Eqiad* est le serveur principal stockant les données “master” et je n’ai pas trouvé d’information sur la manière dont les données sont répliquées avec l’autre serveur “application” (Codfw).

Wikimedia a fait le choix d’ **utiliser énormément la mise en cache pour gérer la charge serveur et répondre rapidement aux requêtes**. Le cache permet de stocker des pages déjà rendues pour les servir plus rapidement (échange du temps de calcul contre de l’espace disque). Ainsi dans le cas de Wikipédia, il est estimé que **90% des requêtes sont servies par des serveurs de cache**. Les serveurs stockent au total 570 To de données, disposent 70 To de RAM et peuvent servir 350 000 requêtes par seconde (en 2019). Ces valeurs sont importantes mais restent relativement faibles par rapport à la charge d’autres sites ou applications (comme les réseaux sociaux ou les plateformes de *streaming* par exemple).

Les serveurs “cache” ne servent pas toutes les pages, mais seulement celles qui sont déjà rendues par les serveurs “application”. Quotidiennement, le cache est réinitialisé et une nouvelle demande de rendu est envoyée à un serveur “application” si la page est demandée par l’utilisateur.

Les médias (audios, images et vidéos) sont stockés dans un serveur de stockage propre **Swift** (OpenStack) pour un total de **428 To de données**. Ces ressources sont partagées avec d’autres projets Wikimedia.

### 3.3 Réseau de distribution de contenu

Wikipédia dispose de son propre **réseau de distribution de contenu** (CDN) pour distribuer les médias et les pages rendues basé sur **Apache Traffic Server**. Le CDN permet de distribuer les données de manière efficace et rapide en fonction de la localisation de l’utilisateur. Il permet

aussi de réduire la charge sur les serveurs de cache et d'application. Il comprend un service de **load balancing** pour répartir la charge entre les différents serveurs et contrôler le trafic (pour les attaques par exemple). Les flux de données entre les différents serveurs sont chiffrés par **TLS** (Transport Layer Security) pour garantir la confidentialité des données.

### 3.4 Vérification des données

L'ajout de nouvelles pages ou de modifications ne sont pas vérifiées directement. Le travail de vérification est effectué par des **bénévoles** qui peuvent être des **patrouilleurs** (qui vérifient les modifications) ou des **administrateurs** (qui peuvent supprimer des pages ou bloquer des utilisateurs). Il existe aussi des **outils automatiques** pour **déetecter les modifications malveillantes ou du vandalisme**.

Ce travail de vérification est essentiel pour garantir la qualité du contenu de Wikipédia mais demande beaucoup de temps et de ressources. Souvent, des messages sont envoyés aux utilisateurs pour leur demander des précisions sur leurs modifications ou bien sont affichés sur les pages pour indiquer que le contenu est à vérifier.

### 3.5 Autres modes d'accès aux données

Wikipédia offre la possibilité d'accéder à ses données libre gratuitement. D'une part, il est ainsi possible de télécharger des **dumps** de données pour les **utiliser hors-ligne ou pour les analyser** (ce que je vais faire). Ces dumps sont disponibles en plusieurs formats (XML, SQL, JSON, etc.) et sont mis à jour régulièrement. Certaines applications (comme **Kiwix**) permettent de télécharger une version de Wikipédia (en ZIM) pour l'**utiliser hors-ligne sur une application** mobile ou un ordinateur afin de rendre cela plus facile qu'avec les fichiers *dumps*.

Sinon, il est aussi possible d'accéder aux données via une **API** (Application Programming Interface) pour **récupérer des données** en temps réel ou pour **automatiser des tâches** (package `wikipedia` en python par exemple). ,

### 3.6 Exemple de parcours de requêtes serveurs depuis le client

#### 3.6.1 Accès à une page

Lorsque vous accédez à une page Wikipédia, vous **accédez en fait à un serveur de cache** qui vous la renvoie (si la page est en cache, plus de 90% des cas). Si elle n'est pas en cache sur le serveur , ce dernier envoie une **requête au serveur “application”** de Wikipédia (Ashburn Virginia, USA le plus souvent) pour chercher une version en cache sur ce dernier sinon le serveur “application” va **lancer MediaWiki pour rendre un HTML** qui sera envoyé à l'utilisateur.

Il est intéressant de noter cette différence en terme de temps de réponse entre une page en cache et une page non en cache en essayant de changer une page dans un langue “relativement” peu parlée sur internet comme le quechua et une autre langue plus parlée comme l’anglais.

### 3.6.2 Mise à jour d'une page

Lorsque vous mettez à jour une page, vous vous connectez directement à un serveur “application”. La première action des serveurs est d’ **invalider le cache** sur tous les serveurs (via *Apache Kafka*). La page corrigée est alors **stockée dans la base de données du serveur “application”**, rendue par et le **cache de ces serveurs est mis à jour**. Cependant, le cache des serveurs “cache” n’est pas mis à jour immédiatement, il le sera au moment de la requête de la page par un nouvel utilisateur sur le serveur (*cf ci-dessus*).

Je n’ai pas trouvé d’information précise sur la manière dont les modifications sont stockées dans la base de données, mais il semblerait que les modifications soient stockées dans une base de données relationnelle (MySQL ou MariaDB). Aussi, je n’ai pas trouvé d’information sur la manière dont Wikimedia vérifie la cohérence des modifications (deux modifications simultanées par exemple).

### 3.6.3 Création d'une nouvelle page

La création d’une nouvelle page est **stockée dans la base de données d’un serveur “application”** et le cache de ce serveur est mis à jour. La page est alors rendue par le serveur “application” et le **cache de ce serveurs est mis à jour**. Cependant, le cache des serveurs “cache” n’est pas mis à jour immédiatement, il le sera au moment de la requête de la page par un nouvel utilisateur (et uniquement sur le serveur “cache” qui reçoit la requête).

## 4 Exemple de manipulation de données *Big Data* sur Wikipédia

En allant sur le site des dumps de Wikipédia, il est possible de télécharger des fichiers XML contenant les pages actuelles de Wikipédia dans une langue. Ces fichiers compressés sont très volumineux (plusieurs Go) et contiennent des informations sur les pages, les utilisateurs, les catégories, les liens, les médias, etc. Il est possible de télécharger des fichiers de **métadonnées** pour avoir des informations sur les pages, les utilisateurs, les catégories, les liens, les médias, etc.

L’objet de cette partie est donc de mettre en pratique de l’utilisation des ressources Wikipédia dans un contexte de Big Data. Les données de Wikipédia (textuelles) étant en accès libre au format **.xml** (qui n’est pas celui utilisé par l’infrastructure WikiMedia) et donc traitable directement en python.

## MediaWiki webrequest flow

Wikipedia, August 2022

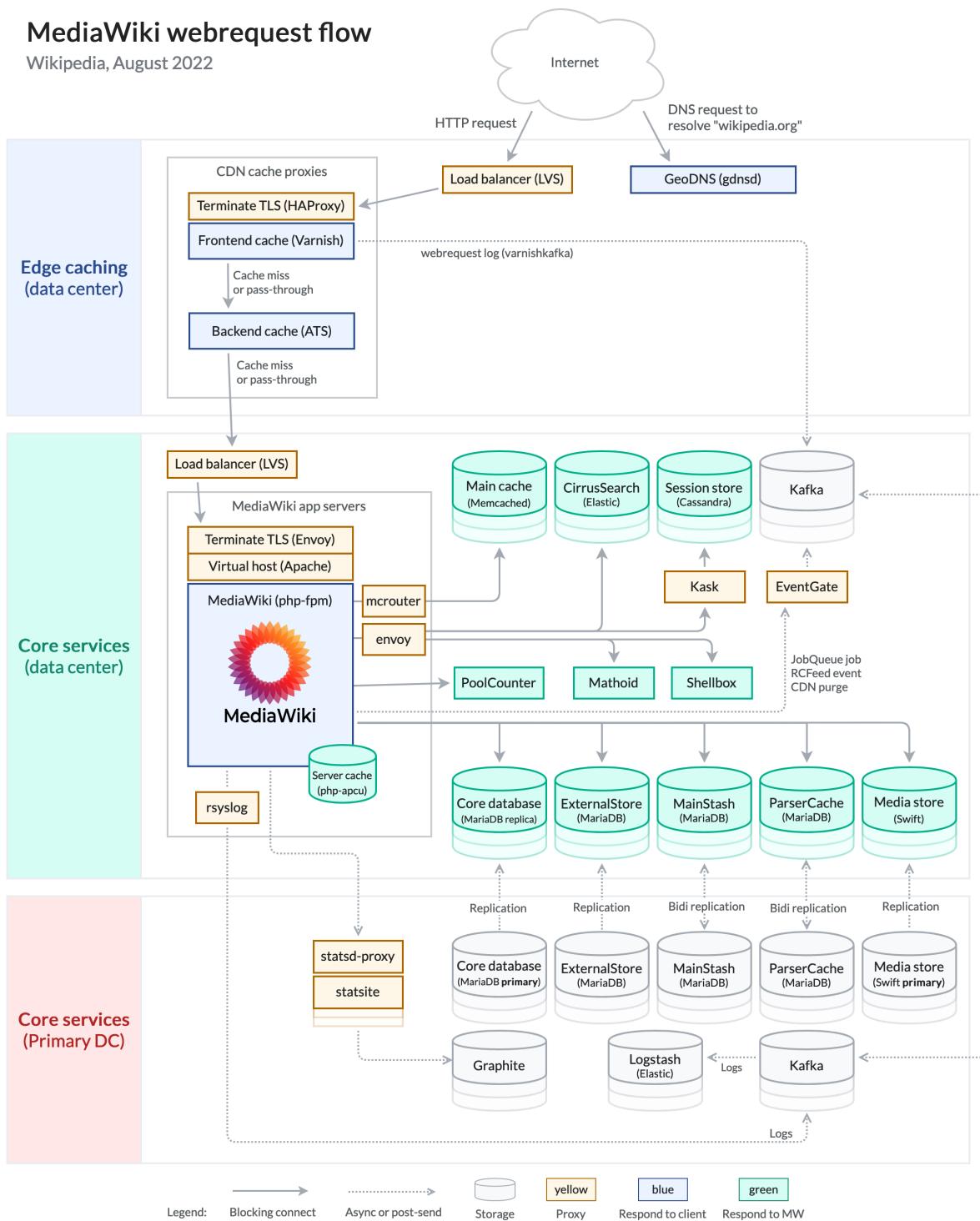


Figure 1: Schéma de l'infrastructure de Wikipédia

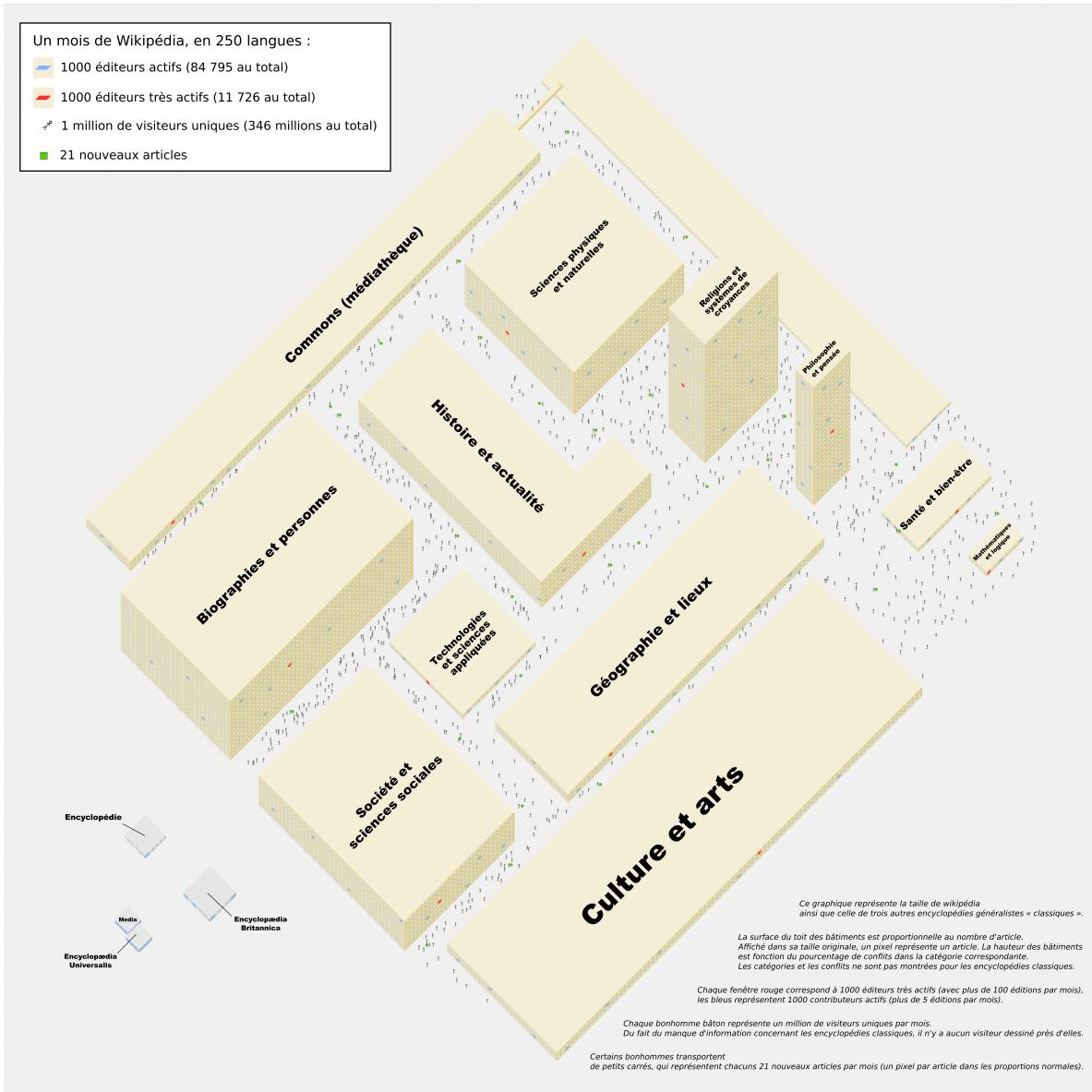


Figure 2: Infographie sur le volume de données de Wikipédia

J'ai préféré travailler sur les données de Wikipédia en galicien pour des raisons de taille de fichier et de facilité de traitement (le français étant trop volumineux pour être traité sur nos machines, et le galicien offrant la possibilité de comprendre à grand traits le texte à des fins de vérification).

J'ai décidé de faire trois choses :

- télécharger les données textuelles de Wikipédia (préalable) en galicien
- une liste de tous les articles de Wikipédia en galicien avec leur titre et leur template (indication sur la catégorie de l'article)
- récupérer le texte de tous les articles de Wikipédia en galicien et les stocker dans une base de données SQL
- faire un interface graphique simple de recherche et d'affichage des articles de Wikipédia en galicien

L'ensemble des fichiers (fonctions appelées, Jupyter Notebook retraçant le tout en détail, architecture de fichier...) sont accessibles dans le [dépôt git](#) associé à ce projet.

## 4.1 Téléchargement des données

Le premier préalable est de télécharger les données de Wikipédia. Pour cela, je suis simplement allés sur le site des dumps de Wikipédia [en galicien](#) et avons téléchargé les fichiers `**wiki-latest-pages-articles.xml.bz2` (étoile selon code de la langue). Les fichiers sont assez volumineux, et le téléchargement peut prendre un certain temps (surtout pour le français ou l'anglais).

Le téléchargement peut se faire en ligne de commande avec la commande `curl` ou `wget`. Par exemple, pour télécharger le fichier `glwiki-latest-pages-articles.xml.bz2` en galicien, on peut utiliser la commande suivante :

```
1 curl https://dumps.wikimedia.org/glwiki/latest/glwiki-latest-pages-articles.xml.bz2 --outp
```

Il faut ensuite décompresser le fichier `.bz2` avec la commande suivante :

```
1 bzip2 -d ./dumps/glwiki-latest-pages-articles.xml.bz2
```

## 4.2 Récupération de la liste des articles depuis le fichier `.xml`

J'ai réalisé un programme sous la forme d'une fonction pour qu'il prenne en entrée le fichier `.xml` téléchargé et qu'il sauvegarde les données désirées dans des fichiers `.csv`.

#### 4.2.1 Ressources

Cette partie est fortement inspirée d'un [tutoriel accessible en ligne](#) et de la [vidéo](#). L'auteur de ce tutoriel a écrit un programme en python qui permet de récupérer la liste des articles de Wikipédia, avec leur titre et leur template à partir du xml et les sauvegarder dans un fichier .csv.

#### 4.2.2 Fonction principale

Comme le fichier .xml est très volumineux, il est préférable de ne pas le charger en mémoire. Il faut le parcourir ligne par ligne pour récupérer les informations intéressantes (en utilisant la librairie `xml.etree.ElementTree` dans mon cas).

J'ai défini une fonction `retrieve_article_info` qui prend en entrée le fichier .xml et qui renvoie la liste des articles, des redirections et des templates dans des fichiers .csv. Le détail de cette fonction est visible dans le fichier `mywiki_funct.py` le [dépôt git](#).

#### 4.2.3 Exécution de la fonction

Les noms de chemin sont assignés avant d'exécuter la fonction `retrieve_article_info` pour le fichier .xml en galicien.

```
1 path_wd = os.path.abspath(os.getcwd())
2 gl_file_path = 'dumps/glwiki-latest-pages-articles.xml'
3
4 file_articles = 'output_csv/articles_gl.csv'
5 file_redicrect = 'output_csv/articles_redirect_gl.csv'
6 file_template = 'output_csv/articles_template_gl.csv'
7
8 retrieve_article_info(path_wd,
9                     gl_file_path,
10                    file_articles,
11                    file_redicrect,
12                    file_template)
```

##### 4.2.3.1 Résultats

Le programme prend un certain temps pour s'exécuter. Voici un extrait du résultat de l'exécution du programme :

```
100,000
200,000
300,000
400,000
Total pages: 409,740
Template pages: 24,441
Article pages: 95,361
Redirect pages: 289,938
Elapsed time: 0:00:28.35
```

Voici les 10 premières lignes du fichier `articles_g1.csv` :

```
id,title,redirect
2,Alimento,
3,Arqueoloxía,
4,Arte,
8,Autor,
9,Autoridade,
11,Acción,
12,Acto,
13,"Actor, actriz",
15,Antropoloxía,
```

Globalement ces fichiers permettent de percevoir l'ensemble des articles du Wikipédia galicien, la partie de ces articles qui renvoient en fait vers d'autres (dans une logique d'une base de données relationnelle) et la liste des templates utilisés pour les articles.

### 4.3 Récupération des textes des articles et stockage

#### 4.3.1 Extraction des textes de articles via `wikiextractor` depuis le fichier `.xml`

A partir du fichier `.xml` téléchargé, j'ai pu extraire les données textuelles des articles de Wikipédia. Pour cela, j'ai eu recours au programme `wikiextractor` disponible sur [ce dépôt github](#). En lançant la commande suivante sur le fichier `.xml` téléchargé, j'ai pu extraire les textes des articles de Wikipédia en galicien dans un dossier `text_g1` :

```
1 python -m wikiextractor.WikiExtractor ./dumps/glwiki-latest-pages-articles.xml.bz2 -o ./te
```

La commande prend quelques dizaines de secondes pour s'exécuter afin d'extraire le texte sur un fichier de plus de 1,3 Go.

```
INFO: Preprocessing './dumps/glwiki-latest-pages-articles.xml.bz2'  
to collect template definitions: this may take some time.  
INFO: Preprocessed 100000 pages  
INFO: Preprocessed 200000 pages  
INFO: Preprocessed 300000 pages  
INFO: Preprocessed 400000 pages  
INFO: Loaded 24441 templates in 87.9s  
INFO: Starting page extraction from ./dumps/glwiki-latest-pages-articles.xml.bz2.  
INFO: Using 7 extract processes.  
INFO: Extracted 100000 articles (1824.7 art/s)  
INFO: Extracted 200000 articles (2209.0 art/s)  
INFO: Finished 7-process extraction of 297324 articles in 145.4s (2045.6 art/s)
```

Le résultat est un dossier `text_gl` contenant des sous-dossiers `AA`, `AB`, `AC` et `AD` contenant chacun des fichiers `wiki_00`, `wiki_01`, ..., `wiki_99` avec tout le texte des articles de Wikipédia séparés par des balises `<doc>` et `</doc>`.

La commande `ls` permet de voir le contenu du dossier `text_gl`.

```
1 ls text_gl
```

Fichiers du dossier `text_gl` :

```
AA  AC  AB  AD
```

```
1 ls text_gl/AA
```

Fichier du sous-dossier `text_gl/AA` :

```
wiki_00  wiki_12  wiki_24  wiki_36  wiki_48  wiki_60  wiki_72  wiki_84  wiki_96  
wiki_01  wiki_13  wiki_25  wiki_37  wiki_49  wiki_61  wiki_73  wiki_85  wiki_97  
wiki_02  wiki_14  wiki_26  wiki_38  wiki_50  wiki_62  wiki_74  wiki_86  wiki_98  
wiki_03  wiki_15  wiki_27  wiki_39  wiki_51  wiki_63  wiki_75  wiki_87  wiki_99  
wiki_04  wiki_16  wiki_28  wiki_40  wiki_52  wiki_64  wiki_76  wiki_88  
wiki_05  wiki_17  wiki_29  wiki_41  wiki_53  wiki_65  wiki_77  wiki_89  
wiki_06  wiki_18  wiki_30  wiki_42  wiki_54  wiki_66  wiki_78  wiki_90  
wiki_07  wiki_19  wiki_31  wiki_43  wiki_55  wiki_67  wiki_79  wiki_91  
wiki_08  wiki_20  wiki_32  wiki_44  wiki_56  wiki_68  wiki_80  wiki_92  
wiki_09  wiki_21  wiki_33  wiki_45  wiki_57  wiki_69  wiki_81  wiki_93  
wiki_10  wiki_22  wiki_34  wiki_46  wiki_58  wiki_70  wiki_82  wiki_94  
wiki_11  wiki_23  wiki_35  wiki_47  wiki_59  wiki_71  wiki_83  wiki_95
```

Chacun des fichiers `wiki_00`, `wiki_01`, ..., `wiki_99` contient le texte de plusieurs articles de Wikipédia. La commande suivante permet de voir le contenu du fichier `wiki_00`.

```
1 cat text_gl/AA/wiki_00
```

Voici un extrait du contenu du fichier `wiki_00` (les balises `<doc>` et `</doc>` indiquent le début et la fin d'un article, les “...” indique que le texte est tronqué pour des raisons de lisibilité, et le texte est en galicien) :

```
<doc id="2" url="https://gl.wikipedia.org/wiki?curid=2" title="Alimento">
Alimento

O alimento é a substancia normalmente comida ou bebida por seres vivos.
O termo alimento inclúe tamén bebidas líquidas.
A comida é a principal fonte de enerxía e nutrición dos animais, e é xeralmente de orixe animal.
Diversas disciplinas encárganse do estudo dos alimentos desde enfoques distintos: a bioloxía
...
</doc>
...
<doc id="187" url="https://gl.wikipedia.org/wiki?curid=187" title="Grupo sanguíneo">
Grupo sanguíneo

...
Frecuencia.
A distribución dos grupos sanguíneos na poboación humana non é uniforme.
O máis común é O+, mentres que o máis infrecuente é AB-.
Ademais, hai variacións na distribución nas distintas poboacións humanas.

</doc>
```

#### 4.3.2 Stockage des fichiers .txt dans une base de données SQL via l'API Python de SQLite

Pour cela, il faut d'abord les séparer en articles individuels puis les stocker en gardant les infos sur les `id`, les `url` et les `titres`. J'ai écrit une suite de fonctions (disponible dans le script `mywiki_funct.py` sur le [dépôt git](#)) en python pour cela.

En appelant la fonction `wrapper_looper_wrapper_splitter_txt` avec les bons chemins, j'ai réussi à obtenir une base de données SQL contenant les articles de Wikipédia en galicien.

```

1 looper_wrapper_splitter_txt(os.path.join(path_wd, 'text', 'text_gl'),
2                               os.path.join(path_wd, 'DB', 'wiki_gl.db'))

```

## 4.4 GUI simple de recherche et d'affichage des articles de Wikipédia en galicien

L'objectif final est de créer une interface graphique simple pour rechercher et afficher les articles de Wikipédia en galicien. Pour cela, j'ai utilisé la librairie `tkinter` de python.

### 4.4.1 Création de la classe WikiSearch

On construit d'abord la classe permettant de formaliser l'interface utilisateur avec une barre de recherche (Github Copilot m'a été d'une grande aide...). Elle inclut une barre de recherche, un bouton pour lancer la recherche, un espace pour afficher les résultats et un scrollbar pour naviguer dans les résultats.

```

1 class WikiSearchGUI:
2     def __init__(self, root):
3         self.root = root
4         self.root.title("Search GUI for WikiGallego")
5         self.root.geometry("1200x600")
6         self.root.config(bg='lightblue') # Set background color to light blue
7
8         self.label = tk.Label(root, text="Enter the name of an article:",
9                               font=("Arial", 14), bg='lightblue')
10        self.label.pack()
11
12        self.entry = tk.Entry(root, font=("Arial", 14))
13        self.entry.pack()
14
15        self.button = tk.Button(root, text="Search",
16                               command=self.search_database, font=("Arial", 14))
17        self.button.pack()
18
19        # Create a scrollbar
20        self.scrollbar = tk.Scrollbar(root)
21        self.scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
22
23        # Create a Text widget with a scrollbar
24        self.result_text = tk.Text(root, wrap=tk.WORD,
25                                  yscrollcommand=self.scrollbar.set,

```

```

26                         font=("Arial", 14))
27             self.result_text.pack(fill=tk.BOTH)
28
29         # Configure the scrollbar
30         self.scrollbar.config(command=self.result_text.yview)
31
32     def search_database(self):
33         title = self.entry.get().lower()
34         conn = sqlite3.connect('DB/wiki_gl.db')
35         cursor = conn.cursor()
36         cursor.execute(f"SELECT title, url, text FROM articles
37                         WHERE LOWER(title) = '{title}'")
38         result = cursor.fetchone()
39         conn.close()
40
41         if result is not None:
42             self.result_text.delete(1.0, tk.END)
43             self.result_text.insert(tk.END, f"Title: {result[0]}\nURL: ")
44             self.result_text.insert(tk.END, f"{result[1]}\n", "link")
45             self.result_text.insert(tk.END, f"Text: {result[2]}")
46             self.result_text.tag_config("link", foreground="blue", underline=1)
47             self.result_text.tag_bind("link", "<Button-1>",
48                                     lambda e: webbrowser.open_new(result[1]))
49         else:
50             self.result_text.delete(1.0, tk.END)
51             self.result_text.insert(tk.END, f"No data found for the title '{title}'")

```

#### 4.4.2 Appel de la classe WikiSearch

On appelle ensuite la classe `WikiSearch` pour créer l'interface graphique.

```

1 root = tk.Tk()
2 app = WikiSearchGUI(root)
3 root.mainloop()

```

#### 4.4.3 Résultat

L'interface graphique affiche d'abord une barre de recherche :

La recherche sur la base de données se fait en utilisant la librairie `sqlite3` de python. La recherche se fait sur le titre de l'article (peu importe les majuscules).

Si un article est trouvé, le titre, l'url et le texte de l'article sont affichés. Si l'utilisateur clique sur l'url, il est redirigé vers la page de l'article sur Wikipédia.

Sinon un message indiquant que l'article n'a pas été trouvé est affiché.

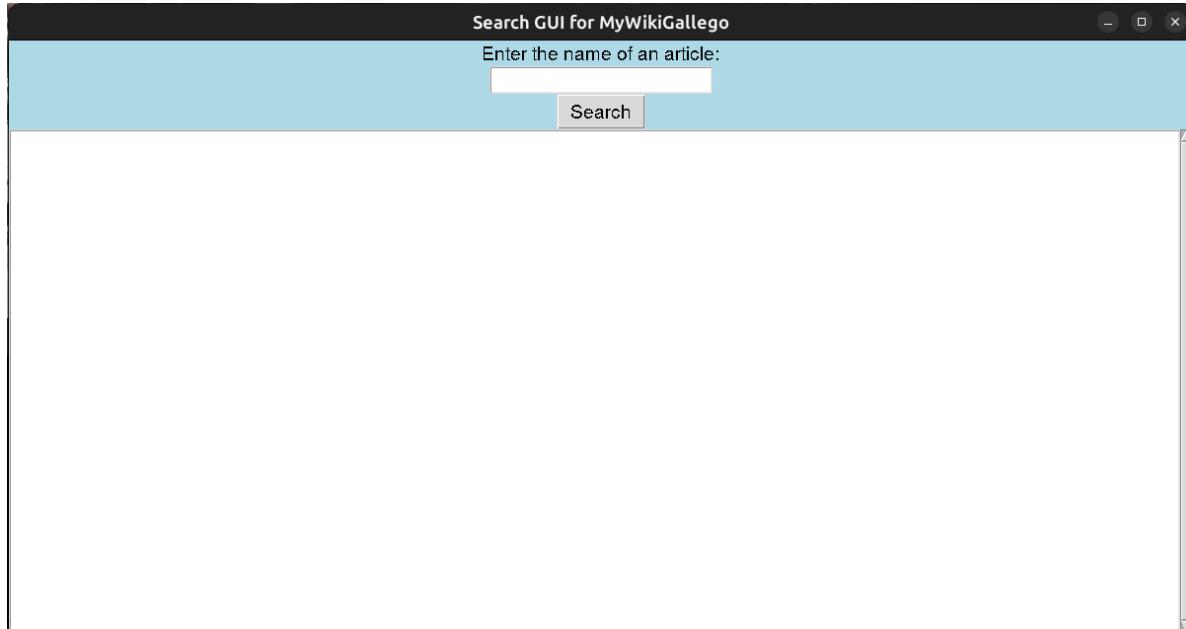


Figure 3: Recherche

#### 4.4.4 Comparaison avec la vraie interface de Wikipédia

Comme il a été dit plus haut, Wikipédia met en cache les pages déjà rendues sous HTML (90% des cas) si elle est disponible, sinon les rends depuis les données de la base MariaDB.

Notre interface simple ne dispose pas de cache, la recherche dans la base de données et le rendu se fait donc à chaque fois que l'utilisateur clique sur le bouton "Search". Aussi, la recherche se fait sur le titre de l'article et non sur le contenu de l'article. Enfin, l'interface graphique est très simple et ne permet pas de faire des recherches avancées ou de naviguer dans les catégories.

## 5 Sources

### 5.1 Articles Wikipédia (inception)

- Article Wikipédia sur Wikipédia
- Article Wikipédia sur les statistiques de Wikipédia

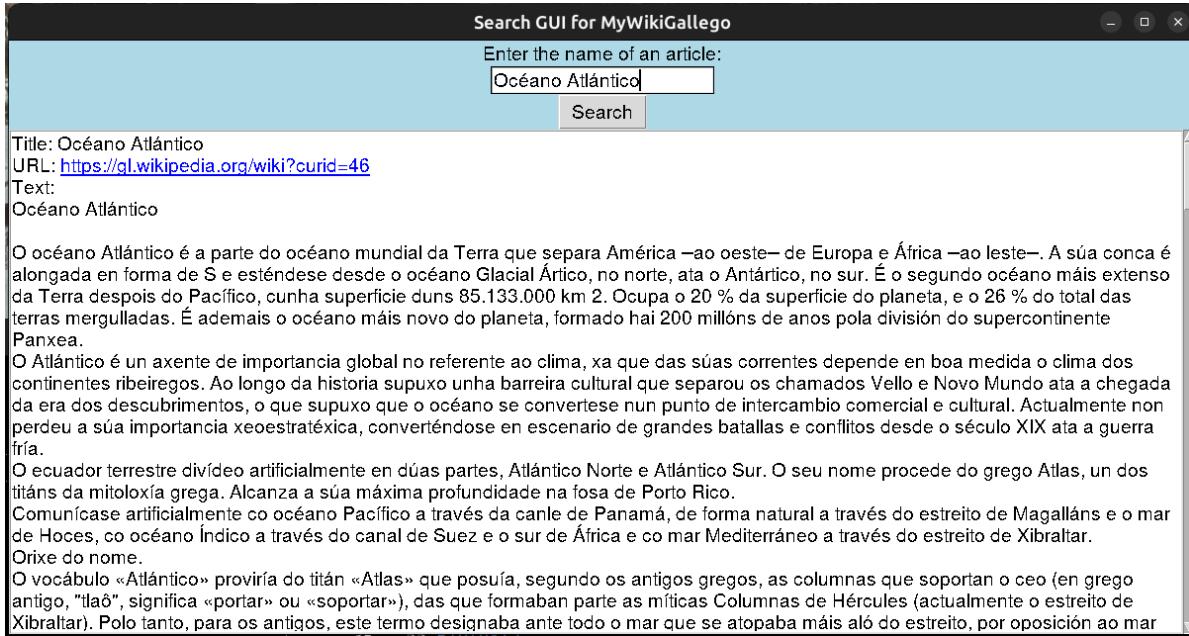


Figure 4: Résultat

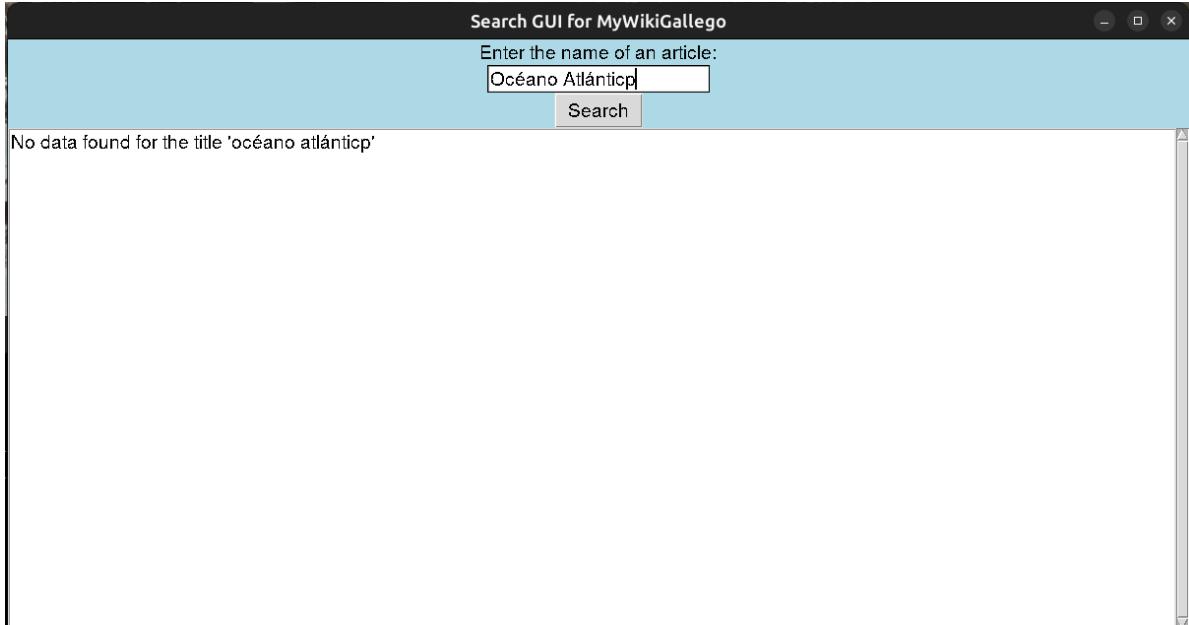


Figure 5: Pas trouvé

- Article Wikipédia sur la Wikimedia Foundation
- Article Wikipédia sur le fonctionnement des serveurs de wikipédia

## 5.2 Sites d'information sur les aspects techniques de wikimédia

- Status des serveurs wikipédia
- Dashboard sur les statistiques de Wikipédia (et d'autres projet wikimédia)
- Documentation technique sur MediaWiki

## 5.3 Télécharger les données de Wikipédia

- Liste des dépôt de données de Wikipédia
- ‘Last dump’ de Wikipédia en français
- Application Kiwik pour lire Wikipédia en hors-ligne

## 5.4 Conférences

- Conférence de la Wikimedia Foundation lors de la 36C3 (conférence internationale sur le *hacking*)

## 5.5 Articles de presse

- Article du Monde sur les manipulations de Wikipédia pendant la présidentielle 2022
- Article de l'Humanité sur les bénévoles de Wikipédia
- Article de l'Humanité sur le fonctionnement économique de Wikipédia
- Article du Monde sur la réouverture de Wikipédia en Turquie

## 5.6 Tutoriels et autres ressources pour extraire les données de Wikipédia

- Tutoriel pour extraire les titres des articles de Wikipédia en .xml
- Vidéo pour extraire les titres des articles de Wikipédia en .xml
- Dépôt github de `wikiextractor`
- Dumps de Wikipédia en galicien
- Wrapper Python de l'API de Wikipédia