

Méthodes algorithmiques pour l'accès à l'information numérique

Gougulle, le moteur de recherche de l'avenir

Université de Paris

A stylized, red, cursive logo for 'Gougulle'. The letters are interconnected in a fluid, handwritten style. The 'G' is large and loops around the 'o', and the 'u' and 'l' are also connected. The word ends with a period.

Logo by Vika bEnJaMIN

Aliaksandr Budzko,
Chawki Chouib,
Benjamin Viau,

21 mars 2021

Table des matières

1	Introduction	3
2	Implémentation	3
2.1	Préparation du corpus	3
2.2	Graphe	5
2.3	Nettoyage du corpus	5
2.4	Dictionnaire	8
2.5	Pagerank	8
2.6	Requête	8
2.7	Serveur	9
2.8	Résumé	10
3	Réponses aux TP's	11
3.1	TP1	11
3.2	TP2	11
3.3	TP3	11

1 Introduction

Mes chers camarades et moi-même eûmes la chance de travailler sur un moteur de recherche pour le Wikipédia français que nous allons vous détailler prestement.

2 Implémentation

Dans cette partie on explique nos choix pour extraire le corpus de travail du dump Wikipédia donné.

2.1 Préparation du corpus

Nous avons eu à notre disposition près de 5 millions de pages Wikipédia sous forme d'un fichier de 25go. On avait comme consigne de travailler sur une partie de cet ensemble de pages uniquement, notamment en choisissant un thème.

Pour procéder, nous avons choisi de se limiter aux pages Wikipédia ayant pour sujet la littérature.

Pour commencer , nous avons parsé le grand fichier XML, tout en faisant attention à ne pas le mettre entièrement en mémoire, afin d'en tirer seulement les informations dont on aurait besoin par la suite.

```

1 <mediawiki>
2   <siteinfo>
3     <sitename>...</sitename>
4     <dbname>...</dbname>
5     <base>...</base>
6     <generator>...</generator>
7     <case>...</case>
8     <namespaces>
9       <namespace>...</namespace>
10    </namespaces>
11  </siteinfo>
12  <page>
13    <title>...</title>
14    <ns>...</ns>
15    <id>...</id>
16    <revision>
17      <id>...</id>
18      <parentid>...</parentid>
19      <timestamp>...</timestamp>
20      <contributor>
21        <username>...</username>
22        <id>...</id>
23      </contributor>
24      <minor/>
25      <model>...</model>
26      <format>...</format>
27      <text>...</text>
28    </page>
29 </mediawiki>

```

Listing 1 – Structure XML avant

Ainsi, en enlevant toutes les balises XML non nécessaires et en filtrant les pages en fonction de leur catégorie, on obtient un nouveau fichier XML qui contient seulement les pages appartenant au domaine de la littérature.

```

1 <mediawiki>
2   <page>
3     <title>page 1</title>
4     <id>1</id>
5     <text>...</text>
6   </page>
7   <page>
8     <title>page 2</title>
9     <id>2</id>
10    <text>...</text>
11  </page>
12 </mediawiki>

```

Listing 2 – Structure XML après

De plus, en même temps que l'on écrivait dans ce fichier, on remplissait une liste **pagelist** ayant la forme (*id, titre, contenu*) et qui a par la suite été sérialisée et sauvegardée sur le disque.

2.2 Graphe

Tout d'abord, nous faisons un premier parcours de la liste de pages pour créer une structure de données contenant toutes les arêtes du graphe (les liens des pages). Puis nous créons simplement le graphe en forme CLI, en prenant soin de ne pas prendre en comptes les arêtes qui pointent en dehors de notre corpus.

2.3 Nettoyage du corpus

Cette partie est consacrée à expliquer comment on a implémenté le passage d'un contenu de départ (avec des éléments de mise en page etc) à un contenu propre et prêt à être parsé.

```
1 <pages>
2 <title>Antoine Meillet</title>
3 <ns>0</ns>
4 <id>3</id>
5 <revision>
6 <id>178204512</id>
7 <parentid>178097574</parentid>
8 <timestamp>2020-12-30T10:12:14Z</timestamp>
9 <contributor>
10 <username>Rovo</username>
11 <id>34820</id>
12 </contributor>
13 <minor/>
14 <model>wikitext</model>
15 <format>text</format>
16 <text bytes="11274" xml:space="preserve">
17 {{Infobox Biographie2|charteslinguiste
18 |nom = Antoine Meillet
19 |nationalité = {{France}}
20 |date de naissance = {{Date de naissance|11|novembre|1866}}
21 |lieu de naissance = [[Moulins (Allier)|Moulins]] ([[Allier (département)|Allier]])
22 |date de décès = {{Date de décès|21|septembre|1936|11|novembre|1866}}
23 |lieu de décès = [[Châteaumeillant]]
24 |région = Linguiste occidental
25 |époque = {{sXX}}
26 |image = Meillet Antoine.jpg
27 |légende =
28 |domaine = [[Linguistique comparée]]
29 |principaux intérêts =
30 |influencé par =
31 |influence de =
32 |idées remarquables = [[épithète homérique]]
33 |œuvres principales = "Introduction à l'étude comparative des langues indo-européennes" ([[1903]])&lt;br /&gt;
34 "Aperçu d'une histoire de la langue grecque" ([[1913]])&lt;br /&gt;
35 "Dictionnaire étymologique de la langue latine" ([[1932]])
36 |adjectifs dérivés =
37 }}
38
39 '''Paul Jules Antoine Meillet''', né le {{Date de naissance|11|novembre|1866}} à [[Moulins (Allier)|Moulins]] ([[Allier]])
40
41 == Biographie ==
42 D'origine bouronnaise, fils d'un notaire de [[Châteaumeillant]] ([[Cher (département)|Cher]]), Antoine Meillet fait
43 Étudiant à la [[faculté des lettres de Paris]] à partir de [[1885]] où il suit notamment les cours de [[Louis Havet]]
44
45 En 1889, il est major de l'[[agrégation de grammaire]]&lt;ref&gt;http://rhe.ish-lyon.cnrs.fr/7q=agregsecondaire_laure
46
47 Il assure à la suite de Saussure le cours de [[grammaire comparée]], qu'il complète à partir de 1894 par une conféren
```

FIGURE 1 – Preview structure de départ

Dans la figure ci-dessus nous pouvons voir un exemple de structure de page dans le fichier wiki.xml.


```

1 <page>
2 <title>Antoine Meillet</title>
3 <id>3</id>
4 <text>
5 Paul Jules Antoine Meillet, né le  à Moulins (Allier) et mort le  à Châteaumeillant (Cher), est le principal linguiste
6
7 == Biographie ==
8 D'origine bourbonnaise, fils d'un notaire de Châteaumeillant (Cher), Antoine Meillet fait ses études secondaires au
9
10 Étudiant à la faculté des lettres de Paris à partir de 1885 où il suit notamment les cours de Louis Havet, il assiste
11
12 En 1889, il est major de l'agrégation de grammaire.
13
14 Il assure à la suite de Saussure le cours de grammaire comparée, qu'il complète à partir de 1894 par une conférence s
15
16 En 1897, il soutient sa thèse pour le doctorat ès lettres (Recherches sur l'emploi du génitif-accusatif en vieux-slav
17
18 Secrétaire de la Société de linguistique de Paris, il est élu à l'Académie des inscriptions et belles-lettres en 1924
19
20 Il a formé toute une génération de linguistes français, parmi lesquels Émile Benveniste, Marcel Cohen, Georges Dumézil
21
22 Il a influencé aussi un certain nombre de linguistes étrangers. Il a également été le premier à identifier le phénomène
23
24 selon le linguiste allemand Walter Porzig, Meillet est un « grand précurseur ». Il montre, par exemple, que, dans les
25
26 L'acte de naissance de la sociolinguistique est signé par Antoine Meillet fondateur de la sociolinguistique qui s'est
27
28 == Études arméniennes ==
29 * 1898, une mission de trois mois dans le Caucase lui permet d'apprendre l'arménien moderne.
30 * 1902, il obtient la chaire d'arménien de l'école des langues orientales.
31 * 1903, nouvelle mission en Arménie russe, il publie son Esquisse d'une grammaire comparée de l'arménien classique, e
32 * 1919, il est cofondateur de la Société des études arméniennes avec Victor Bérard, Charles Diehl, André-Ferdinand He
33 * 1920, le , il crée la Revue des études arméniennes avec Frédéric Macler.
34
35 == Études homériques ==
36 À la Sorbonne, Meillet supervise le travail de Milman Parry. Meillet offre à son étudiant l'opinion, nouvelle à cette
37
38 == Principaux ouvrages ==
39 * Esquisse d'une grammaire comparée de l'arménien classique, 1903.
40 * Introduction à l'étude comparative des langues indo-européennes, 1903 ( éd.), Hachette, Paris, 1912 ( éd.).
41 * Les dialectes indo-européens, 1908.
42 * Aperçu d'une histoire de la langue grecque, 1913.
43 * Altarmenisches Elementarbuch, 1913. Heidelberg (en français : Manuel élémentaire d'Arménien classique, traduction e
44 * Linguistique historique et linguistique générale, 1921 (le tome II est paru en 1936 ; les deux tomes ont été réunis
45 * Les origines indo-européennes des mètres grecs, 1923.
46 * Traité de grammaire comparée des langues classiques, 1924 (avec Joseph Vendryès).
47 * La méthode comparative en linguistique historique, 1925, Oslo, Instituttet for Sammenlignende Kulturforskning (réim
48 * .
49 * Dictionnaire étymologique de la langue latine, 1932 (en collab. Avec Alfred Ernout (1879–1973), éd. augmentée, par

```

FIGURE 3 – Preview structure après nettoyage du text

Dans la figure ci-dessus nous pouvons voir le résultat du nettoyage de la page Wikipédia pour la rendre lisible.

2.3.1 Nettoyage

Le processus de nettoyage prend en entrée la liste des pages et modifie leur contenu de manière suivante :

- suppression des balises "internes" (<>, []...)
- suppression des URL
- suppression des caractères de mise en page (par exemple le "" au début de l'article)
- suppression des sections non pertinentes (par exemple "=== Références ===")
- suppression des espaces blancs
- **tokenisation et lemmatisation** avec Spacy (voir plus bas)
- suppression de la ponctuation
- sauvegarde de la version nettoyée en lettres minuscules

La nouvelle version de la **pagelist** est également sérialisée et stockée sur le disque.

2.3.2 Tokenisation

Pour tokeniser le contenu d'une page, nous avons choisi d'utiliser la bibliothèque Spacy, car malheureusement nltk, étant très bien pour du traitement de texte en anglais, s'est avéré moins efficace en français. Nous appliquons la tokenisation avant de mettre le texte en minuscule et avant d'enlever la ponctuation car Spacy est capable de comprendre le contexte des mots dans une phrase et, grâce à cela, les lemmatiser de manière plus précise.

2.4 Dictionnaire

Pour le dictionnaire nous avons choisi la structure suivante :

```
{mot: ({page: freq}, -)}
```

Pour à la fin retourner :

```
{mot: ({page: TF_norm}, IDF_mot)}
```

Nous initialisons 2 dictionnaires, un **dico-titre** pour les mots des titres et un deuxième **dico-text** pour les mots des textes.

En parcourant chaque page on remplit **dico-text** en incrémentant de 1 la fréquence de chaque mot qu'on rencontre et en ajoutant de nouveaux mots si besoin. On fait de même avec **dico-titre** sauf qu'on incrémente de 100 la fréquence pour donner plus d'importance aux mots des titres.

Nous fusionnons ensuite **dico-titre** avec les 200000 mots les plus fréquents du **dico-text** en obtenons de cette manière notre dictionnaire.

Nous calculons et normalisons ensuite le TF de chaque mot pour chaque page, puis nous calculons le IDF de chaque mot.

Ainsi, nous obtenons notre structure de retour.

2.5 Pagerank

À partir de notre matrice CLI, nous faisons d'une pierre deux coups en calculant le produit $M^t * R$ directement, c'est-à-dire sans passer par le calcul de M^t .

Nous avons choisi un paramètre optionnel pour initialiser le nombre d'itération du produit matriciel, et ainsi contrôler au mieux la précision des probabilités.

Pour le choix de nos α et β , nous avons comparé les ordres de grandeur des pagerank et des $f(d, r)$ sur un ensemble de requêtes aléatoires et avons pris la différence moyenne.

2.6 Requête

La requête de l'utilisateur consiste en un ensemble de mots. Elle est nettoyée par la même fonction qui nettoie le texte des pages pour au final avoir un ensemble de mots compris dans notre dictionnaire de relation mots-pages.

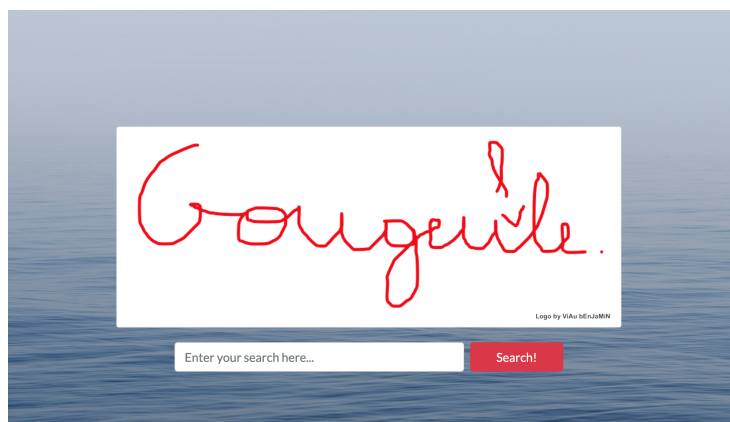
Une fois notre ensemble de mots de la requête prêt, nous utilisons notre dictionnaire de relation mots-pages pour énumérer toutes les pages contenant nos mots de la requête, puis on calcule le score de chacune des pages pour les afficher dans l'ordre décroissant.

2.7 Serveur

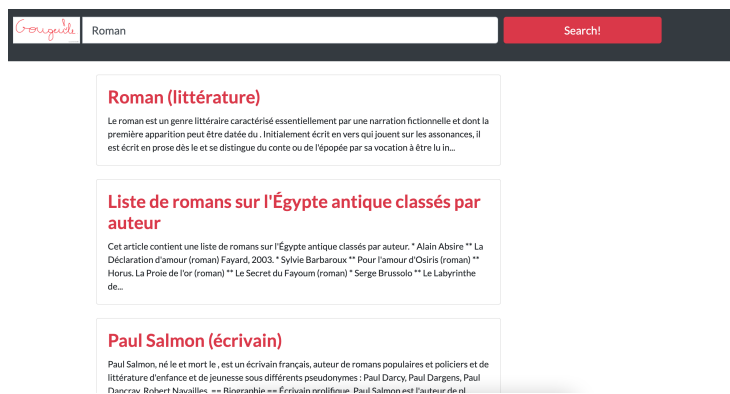
Pour le serveur Web, nous avons décidé d'utiliser Tornado. C'est un framework assez minimaliste pour créer des applications Web interactives de manière intuitive et rapide.

Du côté de l'interface utilisateur, le serveur imite les moteurs de recherche modernes avec une page d'accueil classique et une page d'affichage des résultats. (voir les figures ci-dessous)

Le serveur déséréalise les structures nécessaires au démarrage et ensuite attend de recevoir des requêtes utilisateur.



Page d'accueil



2.8 Résumé

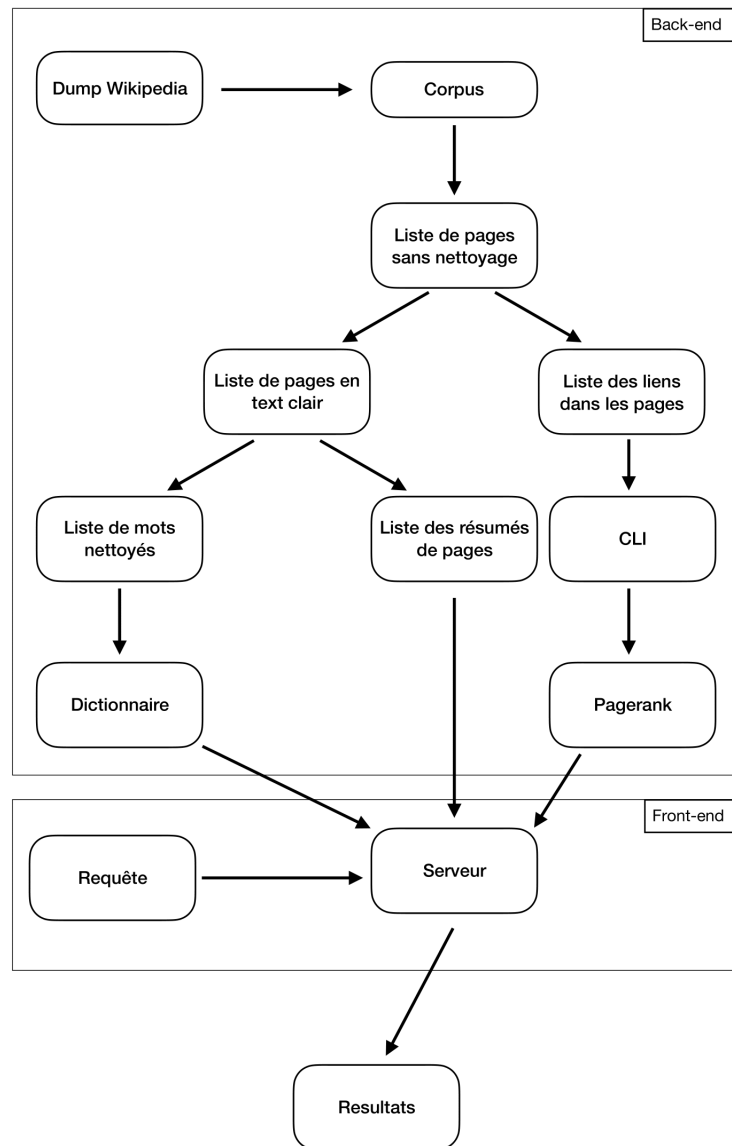


Schéma du projet

3 Réponses aux TP's

Nous avons déjà répondu à la plupart de questions dans les sections précédentes, cette partie contient celles qu'on n'a pas pu y insérer.

3.1 TP1

Exercice 5 Matrices

1)
C = [1,2,3,4,5,6,7]
L = [0,1,3,7,7]
I = [2,0,1,3,1,2,3]

2) Voir le code source.

Exercice 6 Taille des structures

1) Les sommets correspondent aux pages du corpus et les arêtes aux liens. On a 252375 sommets et 2958626 liens.
len(C) = 2958626
len(L) = 252375
len(I) = 2958626

2) On aurait au plus $200 \cdot n$ éléments dans la relation mots-pages. Ce nombre ne dépend pas de m.

Exercice 7 Exploration

5) Si le dump n'était pas disponible, on aurait été obligé de créer un Web crawler. C'est un programme qui parcourrait les pages selon une certaine priorité en se connectant aux serveurs Wikipédia. Le web crawler devrait également respecter les délais entre deux connexions au même serveur pour ne pas être pris pour une attaque DoS et banni d'accès futurs.

3.2 TP2

Voir le code source.

3.3 TP3

Nous avons choisi la méthode "simple" car nous n'avons pas vraiment eu le temps de bien implémenter la méthode WAND.

Exercice 6 Améliorations possibles

- Implémentation de l'algorithme WAND.
- Rajouter l'index positionnel des mots dans les pages afin d'améliorer la précision des résultats.
- Pré-calculer les résultats des requêtes les plus fréquentes.
- Se servir du multi-threading dans nos pré-calculs pour diminuer la complexité en temps.
- Trouver un moyen de parcourir la pagelist sans la charger entièrement en mémoire, notamment avec méthodes de flots d'entrée.