# Web Mining
# Lab 01 - Crawling, indexation et recherche de pages Web

Damien Rochat, Dorian Magnin
Master of Science in Engineering
HES-Source

21 mars 2019

## 1 Introduction

Le but de ce laboratoire est de mettre en place un système d'indexation et de recherche de contenus provenant de pages Web. Nous avons utilisé le logiciel Apache Solr pour l'indexation et développé deux petites applications Java permettant respectivement de crawler et indexer des pages Web et d'exécuter des recherches.

Nous avons choisi de travailler avec le site stackoverflow.com, bien connu des développeurs, permettant de répondre à toutes sortes de questions sur le thème de la programmation informatique. Le but était de récupérer les questions posées sur le site et de proposer une recherche simple par mot clé, celle-ci devant retourner les questions les plus pertinents avec le lien vers la page Web afin d'obtenir d'éventuelles réponses.

## 2 Crawler

Le crawler (développé avec la librairie Java Crawler4j) va utiliser comme racine, la page d'accueil du site (`https://stackoverflow.com/`) et visiter tous les liens en restant sur le domaine, sans limite de profondeur. Ensuite, mêmes si elles sont toutes visitées afin d'obtenir le maximum de contenu, seules les pages « question » (au format `https://stackoverflow.com/questions/[chiffre]/`) sont indexées. En effet, il s'agit des seules pages qui sont intéressantes pour notre application. De plus, le crawler mémorise les pages visitées en ne tenant pas compte des paramètres d'URL afin de ne pas indexer plusieurs fois la même page.

Nous nous sommes très vite heurtés au problème de bannissement du site qui est visiblement très restrictif. Après plusieurs tests, nous avons fini par utiliser un taux de 1 requête par secondes (*politeness* de 1000 ms). Ceci rend l'indexation lente, mais nous ne subissons plus de blocage. Après environ 1 heure, Solr avait indexé plus de 2'600 questions.

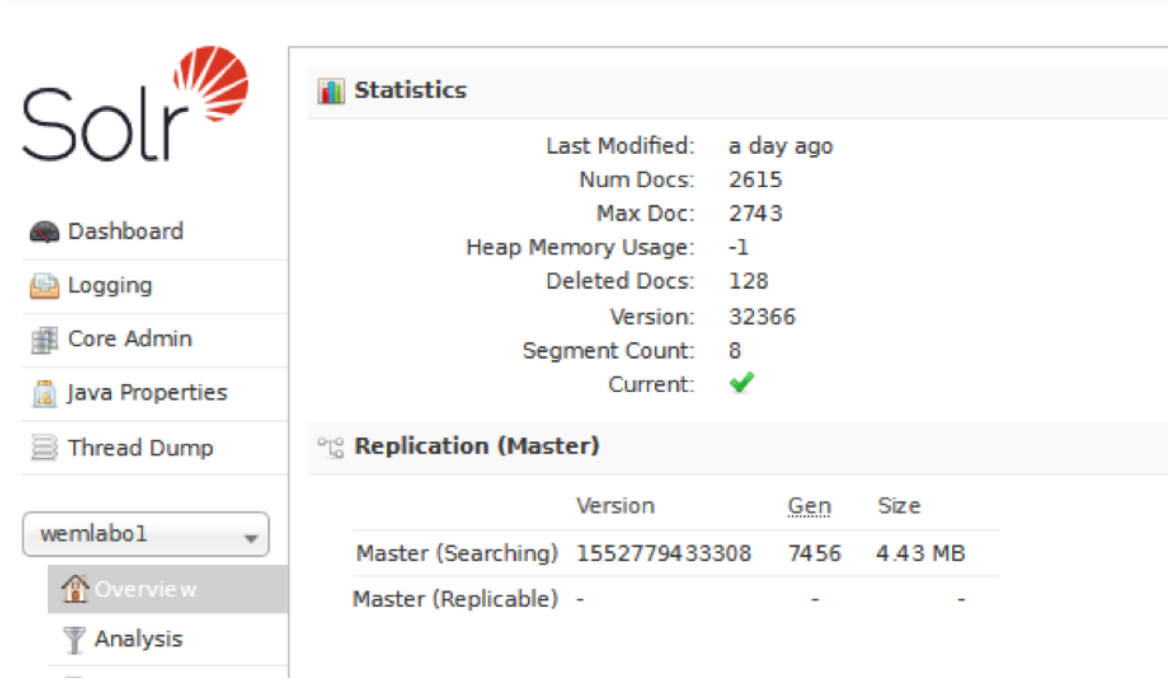


FIGURE 1 – Statistiques du cœur Solr après crawling

Nous n'avons pas rencontré de problème de concurrence. Étant donné le *politeness* assez haut, l'attente entre deux mise à jour de l'index est particulièrement élevée.

## 3 Indexation spécialisée

A l'aide de la librairie jsoup, nous avons récupéré plusieurs informations textuelles dans le code HTML de la page, à savoir le titre de la question, son contenu, les tags que l'utilisateur lui a attribué. Mais nous avons également récupéré la date de publication de la question, si celle-ci a déjà été résolue ou non, ainsi que le nombre de votes positifs qu'elle a reçus de la part de la communauté. Ces quelques champs ne sont pas utiles pour la recherche, mais peuvent l'être pour les utilisateurs effectuant une recherche avec notre application.

En résumé, voici la configuration qui a été définie pour les différents champs du cœur Solr :

```
<field name="id" type="string" indexed="true" stored="true" required="true"
    multiValued="false" />
<field name="url" type="string" indexed="false" />
<field name="title" type="text_general" multiValued="false" />
<field name="content" type="text_general" multiValued="false" />
<field name="tags" type="text_general" multiValued="true" />
<field name="upvotes" type="pint" indexed="false" />
<field name="answered" type="boolean" indexed="false" />
<field name="date" type="pdate" indexed="false" />
```

Les champs « url », « upvotes », « answered » et « date » ne sont pas indexés, mais simplement stockés afin de pouvoir être affichés dans les résutats de la recherche. Ils sont stockés selon leur format correspondant.
Les champs « title » et « content » sont indexés (« text_general » signifie que les textes seront mis en minuscule, tokenizés et que les stop words seront supprimés).
Finalement, les questions peuvent être liées à plusieurs tags. Le champ « tags » est donc un texte,

mais à valeurs multiples. L'id du document est défini en hashant l'url de la page, avec la méthode « hashCode() ».

Ci-dessous, un extrait d'une recherche effectuée depuis l'interface Solr.

```
{
  "responseHeader":{
    "status":0,
    "QTime":34,
    "params":{
      "q":"*:*",
      "_":"1552920712307"}},
  "response":{"numFound":2615,"start":0,"docs":[
      {
        "id":"111253529",
        "url":"https://stackoverflow.com/questions/50720687/in-python-the-path-created-with-os-path-join-is-giving-a-filenotfounderror-er",
        "title":"In python, the path created with os.path.join is giving a FileNotFoundError: [Errno 2] No such file or directory? [duplicate] Ask Question",
        "content":"This question already has an answer here: How can I safely create a nested directory in Python? 25 answers I wanted to save a file uploaded to my database using REST AI
        "tags":["python",
          "flask"],
        "upvotes":1,
        "answered":false,
        "date":"2018-06-06T10:41:40Z",
        "_version_":1628203400908767232},
      {
        "id":"855597461",
        "url":"https://stackoverflow.com/questions/21094467/creation-and-validation-of-directory-using-try-except-or-if-else",
        "title":"Creation and validation of directory using try/except or if else? [duplicate] Ask Question",
        "content":"This question already has an answer here: How can I safely create a nested directory in Python? 25 answers This is only a question regarding which one would be more \"
        "tags":["python",
          "python-2.7"],
        "upvotes":3,
        "answered":true,
        "date":"2014-01-13T14:03:28Z",
        "_version_":1628203409616142336},
      {
        "id":"-1618420902",
        "url":"https://stackoverflow.com/questions/115983/how-can-i-add-an-empty-directory-to-a-git-repository",
        "title":"How can I add an empty directory to a Git repository? Ask Question",
        "content":"How can I add an empty directory (that contains no files) to a Git repository?",
        "tags":["git",
          "directory",
          "git-add"],
        "upvotes":3715,
        "answered":true,
        "date":"2008-09-22T14:41:03Z",
        "_version_":1628203430496436224},
      {
        "id":"-420311103",
        "url":"https://stackoverflow.com/questions/12517451/automatically-creating-directories-with-file-output",
        "title":"Automatically creating directories with file output [duplicate] Ask Question",
        "content":"Possible Duplicate: mkdir -p functionality in python Say I want to make a file: filename = \"/foo/bar/baz.txt\"\n\nwith open(filename, \"w\") as f:\n    f.write(\"FOOB
        "tags":["python",
          "file-io"],
        "upvotes":245,
        "answered":true,
        "date":"2012-09-20T15:06:40Z".
```


FIGURE 2 – Recherche de tous les documents dans Solr

# 4 Recherche

Concernant la recherche, nous avons utilisé (comme pour l'indexation), la librairie solrj. A l'aide de la documentation de Lucene, nous avons pu donner plus de poids au titre, puis au tags et enfin au contenu de la question. Le code se trouve dans la classe Search. Celle-ci contient une petite application permettant d'entrer un mot clé et d'afficher les résultats de la recherche, ordré par pertinence, le tout en mode console.

La première requête Solr que nous avons mise en place était la suivante :

```
(title:[SEARCH])^3 (tags:[SEARCH])^2 (content:[SEARCH])^1
```

Les puissances sont utilisées afin de donner plus ou moins d'importance aux différents champs.
En recherchant « stack trace in Java », chaque mot est recherché par Lucene. Voici le résultat correspondant :

```
====================
Enter something (wildcard allowed): stack trace in Java
Query: 'stack trace in Java'
Documents found: 75
Indexed documents: 2615
Results:
-----
url: https://stackoverflow.com/questions/8804937/java-heap-stack
title: Java heap & stack Ask Question
tags: [java, stack, heap]
upvotes: 2
answered: true
date: Tue Jan 10 14:34:49 CET 2012
score: 35.845078
-----
url: https://stackoverflow.com/questions/6531481/stack-and-heap-in-c-sharp
title: Stack and heap in c# [duplicate] Ask Question
tags: [c#, stack, heap]
upvotes: 12
answered: false
date: Thu Jun 30 07:41:18 CEST 2011
score: 33.597546
-----
url: https://stackoverflow.com/questions/2422252/windows-assembly-heap-and-stack
title: Windows assembly heap and stack? Ask Question
tags: [windows, assembly, stack, heap]
upvotes: 8
answered: true
date: Thu Mar 11 03:05:42 CET 2010
score: 33.475792
-----
```

FIGURE 3 – Résultat de la recherche sans optimisation

Nous avons ensuite amélioré la requête afin de rechercher le terme exacte (la phrase complète) dans le titre, en modifier la requête comme ceci :

```
(title:"[SEARCH]")^3 (tags:[SEARCH])^2 (%s:[SEARCH])^1
```

Voici le résultat correspondant :

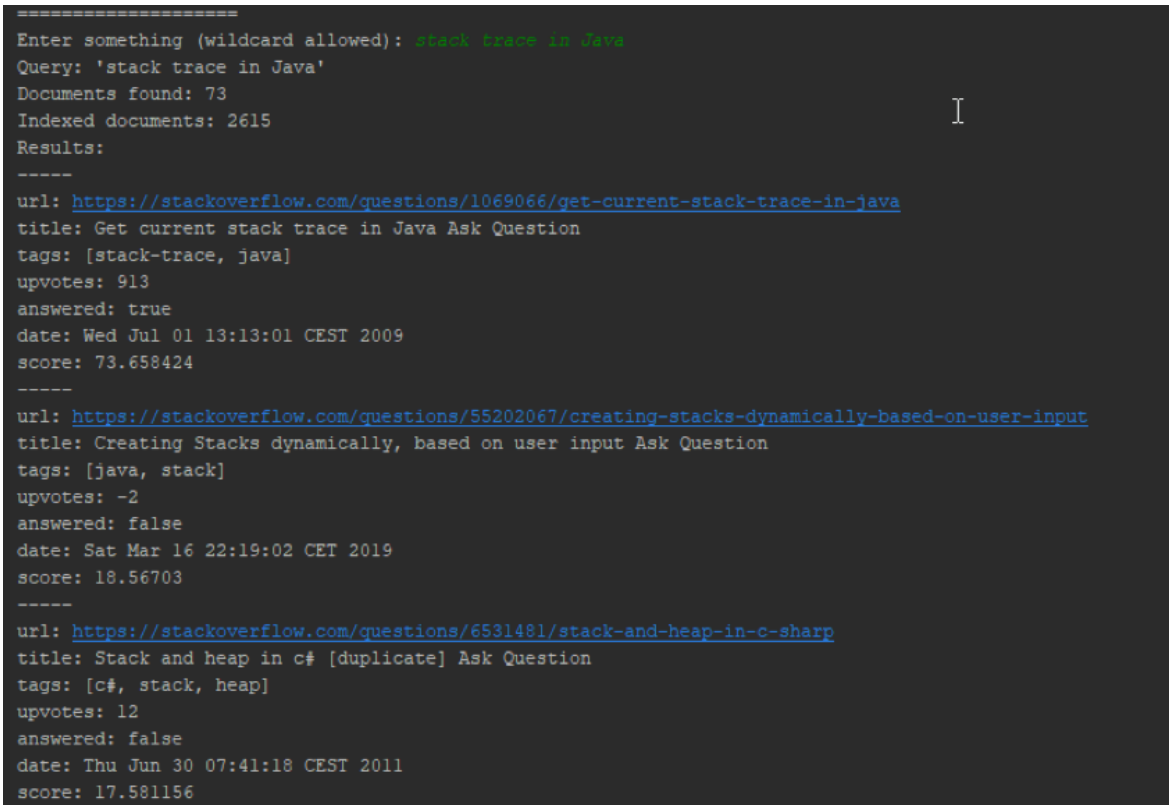


FIGURE 4 – Résultat de la recherche avec terme exacte dans le titre

Cependant, les autres documents se retrouvent fortement défavorisés, leur titre n'étant plus pris en considération s'ils ne possèdent pas le terme exact.

En mixant les deux requêtes, nous sommes arrivés à obtenir des scores adéquats pour chaque question.

```
(title:"[SEARCH]")^4 (title:[SEARCH])^3 (tags:[SEARCH])^2 (%s:[SEARCH])^1
```

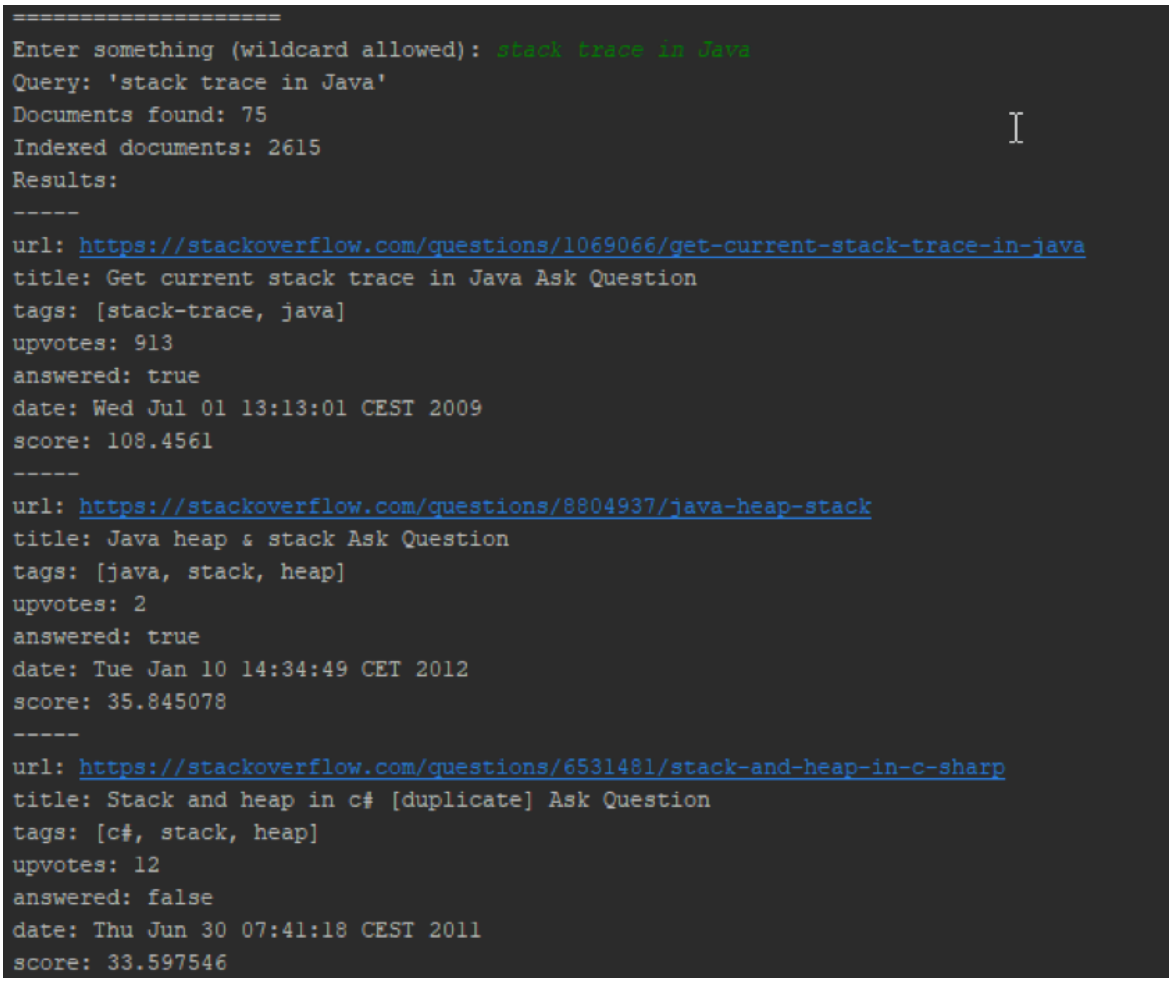


FIGURE 5 – Résultat de la recherche avec optimisation finale

La recherche pourrait encore être améliorée. Sur l'image ci-dessus, on voit bien que les documents retournés de concernent pas Java ou ne concerne pas la stack trace (même si leur score est clairement inférieur à celui du premier document). Ceci peut provenir du fait que finalement peu de documents ont été indexés.

# 5 Questions théoriques

## 5.1 Question 1

**Veuillez expliquer quelle stratégie il faut adopter pour indexer des pages dans plusieurs langues (chaque page est composée d'une seule langue, mais le corpus comporte des pages dans plusieurs langues). A quoi faut-il faire particulièrement attention ? Veuillez expliquer la démarche que vous proposez.**

La langue peut être indiquée dans l'URL ou dans le header de la requête HTTP (« Content-Language ») ou dans le HTML (attribut « lang » de la balise « head »). Dans des cas plus compliqués, il pourrait être possible d'analyser les textes de la page Web afin d'en déduire la langue à l'aide d'une reconnaissance de certains mots-clés spécifiques à chaque langue. Selon le fonctionnement du site Web, il peut être nécessaire d'adapter la requête HTTP envoyée par le crawler en ajouter un header

« Accept-Language », voir même un paramètre POST afin de spécifier au serveur la langue que l'on souhaite.

Une fois celle-ci définie, le crawler va pouvoir filtrer les contenus en fonction des langues à indexer. Deux solutions existent concernant Solr :

— Utiliser plusieurs cœurs, un par langue. Cette approche est simple, mais va utiliser plus d'espace de stockage.
— Utiliser un seul cœur et dupliquer les champs « traduisibles » (e.g. « title_de », « title_fr »). Avec cette méthode, les données communes aux différents languages ne seront pas dupliquées et il n'y a qu'un seul cœur à administrer, indépendemment du nombre de langues.

### 5.2 Question 2

**Solr permet par défaut de faire de la recherche floue (fuzzy search). Veuillez expliquer de quoi il s'agit et comment Solr l'a implémenté. Certains prénoms peuvent avoir beaucoup de variation orthographiques (par exemple Caitlin : Caitilin, Caitlen, Caitlinn, Caitlyn, Caitlyne, Caitlynn, Cateline, Catelinn, Catelyn, Catelynn, Catlain, Catlin, Catline, Catlyn, Catlynn, Kaitlin, Kaitlinn, Kaitlyn, Kaitlynn, Katelin, Katelyn, Katelynn, etc). Est-il possible d'utiliser, tout en gardant une bonne performance, la recherche floue mise à disposition par Solr pour faire une recherche prenant en compte de telles variations ? Sinon quelle(s) alternative(s) voyez-vous, veuillez justifier votre réponse.**

Les requêtes « floues » (FuzzyQuery) permettent de spécifier une distance maximale entre le terme recherché et les termes qui pourront être retournés, selon les algorithmes Damerau-Levenshtein Distance ou Edit Distance. Les résultats ne seront donc pas exactement similaires, mais proches. Ensuite, le score des résultats prendront en compte leur distance respective.
Par exemple, « Caitlin 1 » matchera « Caitilin », « Caitlen », « Caitlinn » ou encore « Caitlyn » qui ont une différence d'un caractère chacun.

Pour des raisons de performance et de précision du résultat, il n'est cependant pas conseillé d'utiliser une distance trop importante (2 au maximum). Le stemming (réduire les termes similaires à un terme commun) peut être utilisé comme alternative dans une majorité des cas.

## 6 Conclusion

Ce travail nous a permis de mettre assez facilement en place un moteur de recherche pour un site tiers. Cependant, nous nous sommes également rendu compte que crawler un site tel que stackoverflow qui contient énormément de contenu et très long, de plus de nouveaux contenus apparaissent et sont modifiés de manière continue ce qui rend l'indexation difficile.

Finalement, même si une simple recherche est aisée à mettre en place, arriver à obtenir les résultats les plus relevants possible nécessite d'implémenter une multitude de techniques (e.g. recherche floue, stemming), de trouver et de pondérer les bons éléments.