Wiki - > https://github.com/USCDataScience/autoextractor/wiki

Clonar

1 - https://github.com/USCDataScience/autoextractor

Instalar

1- openjdk version "1.8.0_292"

```
pedro@pedro:~$ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

2- Apache Maven 3.8.3

```
pedro@pedro:~$ mvn -v
Apache Maven 3.8.3 (ff8e977a158738155dc465c6a97ffaf31982d739)
Maven home: /opt/maven
Java version: 11.0.11, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd
64
Default locale: pt_BR, platform encoding: UTF-8
OS name: "linux", version: "5.11.0-38-generic", arch: "amd64", family: "unix"
```

Build

1 – Vá para a pasta raiz do projeto

2 – mvn clean compile package

Obs: se deu certo, vai aparecer um executável jar
autoext-spark/target/autoext-spark-xx.jar

Execução do Cluster - >
https://github.com/USCDataScience/autoextractor/wiki/Clustering-Tutorial

1- Criar arquivo sequencial, o **-in** é a pasta onde estão todos os html e o **-out** é onde você desejar que apareça esse arquivo sequencial.

sudo java -jar autoext-spark/target/autoext-spark-0.2-SNAPSHOT.jar createseq -in /home/pedro/MPMG/html_saver/ -out /home/pedro/MPMG/web_cluster/autoextractor/sequence/sequencia2

2- Calcular Similaridade estrutural (3 horas), o **-in** é a pasta onde está o arquivo sequencial criado pelo passo 1 e o **-out** é onde você desejar que apareça o resultado.

java -jar autoext-spark/target/autoext-spark-0.2-SNAPSHOT.jar similarity -func structure -in /home/pedro/MPMG/web_cluster/autoextractor/sequence/ -out results/structure -master local

3- Calcular Similaridade css (20 min), o **-in** é a pasta onde está o arquivo sequencial criado pelo passo 1 e o **-out** é onde você desejar que apareça o resultado.

java -jar autoext-spark/target/autoext-spark-0.2-SNAPSHOT.jar similarity -func style -in /home/pedro/MPMG/web_cluster/autoextractor/sequence/ -out results/style -master local

4- Combinar o calculo de similaridades, o **-in1** é onde está o resultado da similaridade estrutural e o **-in2** é onde está o resultado da similaridade css, **-weight** é o peso para a similaridade estrutural, **-out** é onde você quer colocar o resultado da combinação.

java -jar autoext-spark/target/autoext-spark-0.2-SNAPSHOT.jar simcombine -in1 /home/pedro/MPMG/web_cluster/autoextractor/results/structure -in2 /home/pedro/MPMG/web_cluster/autoextractor/results/style -weight 0.4 -out results/combined -master local

5- Fazer o cluster da combinação.

java -jar autoext-spark/target/autoext-spark-0.2-SNAPSHOT.jar sncluster -in /home/pedro/MPMG/web_cluster/autoextractor/results/combined/ -out /home/pedro/MPMG/web_cluster/autoextractor/results/clusters -master local -share 0.8 -sim 0.8

6- Obter o json do clusters.

java -jar autoext-spark/target/autoext-spark-0.2-SNAPSHOT.jar d3export -in /home/pedro/MPMG/web_cluster/autoextractor/results/clusters/ -out /home/pedro/MPMG/web_cluster/autoextractor/results/clusters.d3.json -master local

7- Visualizar o cluster de forma gráfica (bolinhas)

Abra o visuals/webapp/circles-tooltip.html e coloque o json lá.