Polytechnic school of Grenoble-Alpes university

RICM4

Internship report

Student :
Julien Cordat-Auclair

Supervisor :
Sebastien Pittion

13 August 2018

UNIVERSITÉ Grenoble Alpes

Table of content

1 Acknowledgement

Before starting this internship report, I would like to thank Mr. Christian Pomot (director of Com&Net) who accepted to welcome me into his company and who taught me a lot during these 12 weeks by transforming this professional experience into a moment both extremely rewarding and very profitable. He was always available and at listening when I encountered any kind of difficulty, and talking with him allowed me to find a logical solution to each of them.

I also thank my supervisor who has regularly assisted me throughout this period with a lot of pedagogy and who has always been available in case of problem. Finally, I would like to thank all the company's employees for their feedback as well as Théo Echevet (student in RICM4) who has helped me several times, without forgetting Mr. Christophe Brouard (researcher at the Laboratoire Informatique of Grenoble) for his help during this project.

2 Introduction

2.1 Context

From May 21 to August 13, I did an internship at COM&NET. This company specialized in the web allows its customers to develop their websites while optimizing their SEO. During this period, I worked in the company's offices in Grenoble and was accompanied by **M. Christophe Pomot** (director), two employees, an alternate and **Théo Echevet**. The company manager provided me clear and precise specifications for the project he had assigned to me, the main requirement being the use of **M. Christophe Brouard**'s algorithm.

2.2 Mission

During this internship, I had to set up a service to submit recommendations to a user regarding the referencing of his web page for a given request. The user therefore provides the URL of his page as well as the keywords for which he wants it to be referenced at the best and the service must send him a set of information helping him to achieve this objective. This application also uses the machine learning algorithms developed by **M. Christophe Brouard** called `Echo`. The project is therefore distinguished in three specific steps: data collection from the web, machine learning from it and then displaying the results returned by `Echo`.

The project therefore raises back-end and front-end issues, establishing a rich and complete field of study, especially since multiple languages, frameworks and libraries have been used: Python, PHP, HTML, CSS, JavaScript and Scrapy (*scrapy.org*), Selenium (*www.seleniumhq.org*), Symfony (*symfony.com*), jQuery (*jquery.com*) or the template engine Twig (*twig.symfony.com*). In addition, it is important to note that the service I have implemented will be used later by the company to facilitate SEO tasks, both for customers and employees specialized in this field who will then have an additional tool at their disposal to perform this important work when it comes to establishing an effective digital strategy.

Finally, the efficiency of this service must be optimal in order to be able to propose modifications that will allow the sites analysed to gain as many places as possible: it should be noted that only the 3 best positioned pages attract the attention of users, the others going almost unnoticed (only 10% of attention is given to them). The problem of referencing is therefore essential for any person or company wishing to stand out from the others and the interest shown in it can only increase over time.

Here are two tables to illustrate this project; the first presents the parameters entered by the user and the second corresponds to the results that will be returned to him:

user's URL	http://com-et-net.com
keywords	stratégie numérique grenoble

`EchoBT`	référencement_title, agence_title, stratégie_alt, isère_strong ...
`EchoPos`	14
`EchoV`	0.76, 0.63

All these information will be explained later.

2.3 Plan

As stated earlier, three obvious steps have emerged during this project and will form the outline of this report. As a reminder, the first was a web scraping step (data collection from the internet), then came the machine learning (processing of these data) to finally give way to UX Design (optimization of the display of results). However, a presentation of the company in which I have worked is necessary before looking at the work done over the past three months.

3 Com&Net

3.1 Presentation of the company

Com&Net was called Mountain&Net until 2016. Founded in 2003 by **M. Christian Pomot**, this company is focused on the creation of web tools specific to tourism professionals, particularly in the Vercors region in which it was previously located. Today, Com&Net has strengthened its presence in Grenoble by setting up offices there, but it still maintains an antenna in Villard-de-Lans to be able to remain close to its former customers. It is indeed in this city that the company was initially created and it is here that it has been able to develop.

The company thus supports the projects and the presence on the internet of various tourist structures such as hotels, resorts or other activities. Its main assets are obviously the mastery of the IT field but also an excellent approach to the various issues related to tourism. It offers different types of services such as website creation, SEO optimization, promotion and application development.

3.2 Website and application development

The company masters a wide range of frameworks in order to best adapt to the needs of its customers. Symfony (a PHP framework allowing a structured and fast development) and Copix are the most used in the case of large projects. Thelia (an Open Source solution that appeared in 2006) is also a framework that employees are able to use when setting up an e-commerce site. Thus, the company develops many tourist sites but also applications such as GTE which allows the traceability of explosives for winter sports resorts.

In addition to their creation, Com&Net also hosts and maintains the websites. Employees are always available to answer any questions from customers and are ready to solve any problems at any time.

3.3 Digital strategy

Com&Net is also a company specialized in digital strategy, i.e. the different methods used to optimize the visibility of its clients' websites in search engine results. It is partly with this in mind that a partnership was set up with the Grenoble Alpes university in 2015 to be able to use the machine learning algorithms developed by **M. Christophe Brouard**.

3.4 Organisation

Com&Net is a SARL (Limited Liability Company), which means that there is a single partner, in this case **M. Christian Pomot**. Three people are employed in the company and currently accompanied by an alternate (see *appendix 1*). Two sectors are clearly distinguished within it: the first is dedicated to the development of websites and applications, the second is dedicated to digital strategy and web marketing.

3.5 Sustainable development

To reduce its carbon footprint, COM&NET has chosen to have its various servers hosted by PHPNET. The eco-responsible approaches that this Grenoble company offers are as follows:

- the use of SSD servers offering optimal reliability and a significant reduction in power consumption (nearly 98% less energy than traditionally)
- the donation of old computer equipment to the association *Ulisse Solidura* with a view to recycling through their workshop named *DEEE* (Waste Electrical and Electronic Equipment)
- the use of so-called Free Cooling technologies designating natural cooling and thus allowing the company's datacenter to be cooled by outside air if its temperature is lower than that of the inside (therefore very practical in winter)
- the connection of the data centers to the Grenoble power grid GEG (green electricity)
- hosting of servers for associations such as *Les Jardins de la Solidarité*, an integration project combining organic market gardening, a nursery and green spaces

In addition, COM&NET uses TERRA computers to reduce power consumption due to computer equipment. These are certified by the ENERGY STAR 5.0 and therefore have low energy consumption. The company therefore reduces energy consumption in addition to greenhouse gas emissions. TERRA also optimizes its computers a little more over time (about 35% reduction in consumption between current devices and devices that are 3 years old). The screens are also chosen by this brand, implying an even greater overall reduction in energy consumption.

Finally, it is important to note that **Christian Pomot** owns an electric car and is far from insensitive to sustainable development. Trainees and alternates who have had the opportunity to work in the company's offices in Grenoble have always traveled by tram or on foot.

4 First part: web scraping

Before any development, a diagram representing the general architecture of this step and the links between the different files that compose it is available (see appendix 2*).*

4.1 General overview

The first phase of this project focused on data recovery from the web. As the internship is focused on the theme of machine learning, it seems essential to be able to provide elements to the machine learning algorithm in order for it to work. As the purpose of the service is to optimize the referencing of a web page, the idea is to focus on public data related to this domain. This step consists in collecting the content of the tags that weigh on the balance of the referencing of a web page, and this for a significant number of sites displayed following a Google request (this request being chosen by the user). We then speak of SEO On-Site (SEO stands for Search Engine Optimization, On-Site designating all the SEO optimization actions that take place on the website and its contents). The positioning of the user's site cannot be optimized in the most efficient way thanks to this method since the Off-Site SEO (everything related to the very structure of the site) is not supported by the service. However, it is an improvement that the director of Com&Net is strongly considering as part of the continuation of this project in order to obtain a complete but above all very efficient tool.

To do this web scraping work, **M. Christian Pomot** insisted that I use the language Python as well as the Selenium and Scrapy frameworks; the first one simulates a web browser and the "human" interactions that go with it (like a mouse click or a pressed keyboard key) and the second one makes it possible to create indexing robots capable of performing web scraping tasks (ie. retrieve data HTML). Their combination therefore seems perfectly adapted to the project.

4.2 Description of the work performed

4.2.1 Retrieving URLs

First, I used Selenium to be able to generate a Google query based on the keywords entered by the user. These keywords, as well as the URL of the latter's web page, are actually stored in a text file (with other parameters) and the web scraping engine is able to extract them. In this way, the retrieval of URLs and the entire process that follows will be adapted to the user's request. The product text file is interpreted as follows:

1st line	user's URL
2nd line	number of pages to be studied
3rd line	degree of pertinence
following lines	keywords

The idea is to retrieve the URLs present on the results page of the user request in order to extract the data that interests us. A Google search link can be written: `https://www.google.com/search?q=k` You simply have to change the keywords of this link to those chosen by the user to be able to display the desired results page using the functions offered by Selenium.

In addition, Google displays 10 results per query by default. This value can be modified (up to 100) and it was interesting to be able to vary this number, which determines the quantity of URLs, and therefore potential data, to be analyzed later. I then added the suffix `&num=100` to the search link to display 100 results. And if you want to retrieve more than 100 URLs, then the script is able to make a first request where it looks for the first 100 and then generate a new request with the suffix `&sa=N` telling Google to display the second page. Then, it is necessary to identify in the source code of the results page the tags that identify the presence of a link to a web page. Again, a function of texttsf Selenium allows to retrieve the content of the identified tag. This is how the script is able to return a list of URLs classified by position from a set of keywords and a given number of results.

4.2.2 HTML data collection

Secondly, Scrapy helped me to save the SEO data for each of the URLs I had previously found. Once the script involving Selenium has been executed, it is sufficient to provide the list of URLs to Scrapy so that the latter can analyze them. So during this step I specified the different important tags for SEO in order to extract the content for each web page previously returned. Thus, the tags *title*, *h1*, *h2*, *h3*, *strong*, *a*, *p* and the attribute *alt* of the tag *img* and the words appearing in the URL of the site (after the domain name) were retained. According to **M. Christian Pomot** and the majority of studies published on the Internet, all these values are the ones that count for the referencing of a web page. However, Google does not confirm the veracity of these assumptions. On the other hand, it should be noted that the content that is part of the SEO On-Site changes regularly as evidenced by the tag *keywords* which was essential a few years ago but no longer has any impact on SEO today. And it is very easy to adapt to these variations through the script described here since it is enough to manually change the tags to study.

Scrapy finally allows to send back a JSON file containing all the recovered data: it is this file that will later make the link with the machine learning step. This framework is very fast because it works asynchronously. However, this is a problem here since we necessarily lose the position given by the index of the page analyzed in the table of URLs previously returned. So I had to explicitly assign a position to each page to keep this information provided by Selenium.

4.2.3 Link with machine learning

Once the data has been retrieved, the idea is to be able to transmit it in any way to the machine learning algorithm. Knowing that this second major step (presented later) is coded in PHP within a Symfony project, it is essential to be able to provide from Python data that will be readable not only by PHP but also by the machine learning algorithm. The PHP language is able to read a JSON file, so no real modification is necessary on this point (except for some encoding problems) since it is the output format of the Python script. However, the algorithm developed by **Mr. Christophe Brouard** is not able to read any form of punctuation, accentuation or double, triple spaces, etc. It is therefore necessary to ensure that there is none as soon as possible. Before moving on to the next step, a complete cleaning of the data is therefore carried out.

5 Second part: machine learning

Before any development, a diagram representing the general architecture of this step as well as the links between the different files of which it is composed is available (see appendix 3).

5.1 The Echo algorithm

As stated above, this internship revolves around the theme of machine learning since the most important initial specification was the use of the machine learning algorithm (named `Echo`) developed by **M. Christophe Brouard**. In fact, in this project, three algorithms with specific actions and derived from `Echo` were used: `EchoBT`, `EchoV` and `EchoPos`.

- `EchoBT` (*BT* meaning *best-terms*) allows, given a set of words each accompanied by an identifier and a relevance index, to determine which terms appear to be the best ones in order to improve referencing. Based on what was done in the previous step, the words will be the terms contained in each retrieved tag, the identifier will correspond to the name of the site associated with the current word and the relevance index will be a constant that will determine up to which position the sites are judged to be well referenced (for example, we can estimate that the 20 best referenced sites are relevant, so these sites will have a relevance index of 1 and all others will have a relevance index of 2 which corresponds to *non relevant*). Thanks to these parameters, the algorithm will be able, via machine learning methods, to identify which terms are important for SEO by finding those that appear in relevant sites but not (or only slightly) in irrelevant sites. Note that `EchoBT` returns 30 terms ranked in descending order of score (from most effective to least effective), but this value can be modified to obtain more.

- `EchoV` (*V* meaning *verified*) allows with the same parameters as those of `EchoBT` to provide a performance rate representing the quality of the results returned by `EchoBT`. Indeed, it can sometimes be difficult to isolate effective terms for SEO (for example, if all the terms provided to `EchoBT` are identical, then the algorithm will not be able to find a term that appears in one of the relevant sites but is not present in an irrelevant site) and the performance rate will then be very low. Conversely, it may also be easy to know which term is *SEO-efficient* (for example, if all relevant sites contain a common term and that term never appears in one of the sites deemed irrelevant), leading to a high performance rate.

- `EchoPos` (*Pos* meaning *position*) allows from the same parameters accompanied in addition to the content of the user's site as well as the keywords composing the Google query to estimate the position of the page provided for this query. In addition, a confidence rate is calculated to determine whether or not to rely on this result.

5.2 Description of the work performed

5.2.1 Architecture of the Symfony project

It is from this stage of the project that the framework PHP Symfony is used. In addition to linking data recovery to the display of results, the algorithms Echo must be called during the recovery. So it's probably the most important of the three. Symfony imposes a very particular architecture and allows to develop in MVC (*Model-View-Controller*) in a fast, clean and efficient way. To make the most of this framework, I had to organize my project as follows:

- a first script (called *controller* under Symfony) allows to recover the data: it then makes the link with the first step. In addition, it also generates the text file that contains the parameters of the study to be conducted and is used by the web scraping engine.
- a second controller allows to execute the machine learning algorithms with these data: this is the second step. This is essential because it is at the origin of obtaining recommendations regarding content.
- a set of files HTML, CSS and JavaScript allows to display the returned results thanks to jQuery as well as successive calls Ajax (Ajax means *Asynchronous JavaScript + XML* and allows to make requests HTTP into JavaScript). This is the third and final step and requires careful consideration because it requires displaying a lot of data while making it easier for the user to read.

5.2.2 Data transmission

The algorithms used EchoBT, EchoV and EchoPos need data to be able to work. The parameters provided by the user (keywords forming the Google query and page URL) are written in a text file and accompanied by the number of results desired and the degree of relevance. This configuration file is then sent to the script Python which will be executed. However, once the scraping data is retrieved from the Symfony project, prior processing is required before calling the algorithms. Indeed, the terms transmitted to the latter must keep the information related to the tag to which they are associated, otherwise terms of the tag *title* could for example be compared to terms of the tag *h3* and this would not make sense. That's why a suffix "_TITLE" is added to a term in a tag *title*, "_H3" for a term in a tag *h3*, etc. It will therefore be in this format that the data will be transmitted to the algorithms Echo. In addition, the contents of the relevant pages are duplicated a number of times and according to their actual position in the Google results page in order to give them more importance during the machine learning process.

5.2.3 Echo calling and data recovery

Each algorithm runs on a local TCP server with a different port number. Thus, it is enough to connect to the right port and transmit the necessary information to be able to execute the algorithm that interests us and retrieve the results provided.

Once the algorithms have analyzed the data, the results are returned to the PHP script. A function then allows you to sort the best terms provided by `EchoBT` both by tag importance (for example, the tag *title* is more important to SEO than the tag *strong*) and by score. This function also allows you to limit the number of terms to be displayed for the user; for example, you do not want to have 100 terms in the tag *h3*. It will be these pre-processed data that will be posted later.

6 Third part: results display

Before any development, screenshots of the final display are available (see appendices 4, 5, 6 and 7).

6.1 General overview

The last step of this project is dedicated to displaying the results previously returned by `Echo`. It implements issues related to UX Design (for *User Experience Design*), the field of computing which consists in designing a website in the best possible way so that its use is optimal. Initially, the basic criteria for my project and for the user were as follows:

- he must be able to enter keywords (which form the Google query)
- he must be able to enter an URL (the one of the page he wants to optimize)
- he must be able to launch the study
- he must be able to visualize the results of this study
- he must be able to understand these results in order to benefit from them

6.2 Description of the work performed

6.2.1 Data recovery

Before wanting to process the display of results, it is necessary to be able to appropriate them. The link with the previous step is then made in jQuery thanks to Ajax calls which allow to retrieve the information returned by `Echo`. This method also allows each of the steps to be performed one after the other once the user decides to launch the study. This makes it easy to identify any problem if an error occurs during the process. Finally, the use of this structure ensures an optimal execution speed and therefore a minimum waiting time for the user since the different pages (loading and results) are loaded on site.

6.2.2 Results, display and user interface

The third step focuses on optimizing the display of the results returned by `Echo` following learning from a data set provided by the script Python; these are diverse and varied and are composed of the following elements:

- an integer with a percentage returned by `EchoPos` : they correspond respectively to the estimated position of the user's page in Google results for the given query and the confidence rate that can be given to this result.

- a percentage returned by `EchoV` : it corresponds to the performance rate of `EchoBT`. This allows the user to know how confident he or she can be in the results provided by the user.
- a set of n terms of the form"word1_BALISE" where *word1* corresponds to a word considered important with regard to SEO by `EchoBT` and *BALISE* refers to the tag in which this term is located. The whole n can be modified in the code.

I have decided to split the display into two distinct parts: at the very top of the page, the user will see first a section dedicated to general information about the study he has requested (see annex). This includes a summary of his research, the estimated position of his page, links to the best referenced sites for the given request without forgetting two graphs representing the percentages mentioned above. Tooltips accompany them in order to clearly translate their function.

Then, a second section displays the best terms returned by `EchoBT`. Each term is stored in a sub-section dedicated to the tag to which it is associated, and all terms present within the same sub-section are classified by decreasing order of score (a color code is used to reflect this). In addition, the subparts are displayed in order of importance of the corresponding tag: the terms of the tag *title* appear at the top because it is the most important for SEO, but those of the attribute *alt* appear lower because it is considered less effective.

6.2.3 Additional features

Each subpart is accompanied by a text field in which the content of the page to be optimized is written and which is associated with the corresponding tag. This field can be modified by the user: the idea is to write new relevant content using terms indicated above which, according to `Echo`, allow for better referencing. Each tag has its own field that can be freely modified. Once the modifications have been made, the user then gains access to two buttons at the bottom of the page: the first allows the position of the page to be recalculated with the modifications made and the second allows the changes to be exported as a JSON file.

7 Summary

7.1 Analysis of the work carried out

This project was complete and required a lot of thought with regard to its organization. First, it was necessary to code a web scraping engine using Scrapy and Selenium. This one is able to retrieve the HTML data contained in specific tags of the best referenced web pages for a given request; it was mainly intended to provide the necessary and cleaned data (accents, punctuation and unnecessary spaces) to `Echo` for machine learning. This central step of the internship allowed me to take charge of the PHP Symfony framework and I had to, during this one, find a way to make the link with the previous one while keeping the information related to the tag associated with each term. Finally, many issues related to the display of results and its ergonomics were raised, but this UX Design step finally resulted in something complete, clean and easy to read. Moreover, it was during this one that I was able to discover jQuery as well as the different Ajax calls, while using the classic web languages.

7.2 To go further

The tool I was able to develop during this internship is not yet complete and some points could be improved:

- for the moment, the application is dedicated to SEO On-Site (content) but does not take into account SEO Off-Site (structure). Thus, the optimization is not total since the proposed changes only concern a part of the SEO. Moreover, in the current state of application, the position estimate is an estimate based only on content and not on structure, so it is not representative of reality.
- it could be interesting to allow the user to modify the number of pages to be studied and the degree of relevance (i.e. how far are the referenced sites considered relevant to SEO). We realize that in order to obtain consistent recommendations (and therefore a very high performance rate), these parameters must be adapted to the given request and will therefore never be identical. Currently, these values are set in the code (200 results and 30 relevant ones) so that the returned results are interesting for a maximum number of queries, but one can imagine wanting to let the user choose them himself. It will then be sufficient for him to check that the performance rate is high to know whether or not the parameters he has chosen are good. We can even think of a system that automatically calculates the best possible parameters for the keywords provided by seeking to achieve a 100% performance rate.
- some of the terms returned by `EchoBT` are inconsistent (about 1 in 5). This is an easily justifiable phenomenon; in the case of general requests where many pages seek to stand out, the contents of these pages will be more or less similar. As `EchoBT` searches for terms that allow you to stand out, it will not return these keywords there but rather terms that

appear several times and in relevant pages without them appearing in irrelevant pages. These are generally proper names and more precisely site names.

Example: we want to optimize the company's page COM&NET dedicated to web marketing for research "grenoble stratégie digitale". EchoBT then returns inconsistent terms such as"oxiwiz","emalaya" etc. In fact, you realize from the Google results page that the first 50 sites include the searched keywords in their titles. Knowing that the first 30 sites are considered relevant (and the rest irrelevant), the machine learning algorithm will ignore these terms and return words that appear several times in the relevant pages without appearing in the irrelevant ones instead. This is how we end up with inconsistent results:' 'oxiwiz" and"emalaya" are company names, and they are returned because in addition to being well referenced pages, an article appearing in the first results quotes the latter. These are therefore terms that are repeated in relevant pages but not in irrelevant ones.

To avoid this kind of problem, I added in my code and at the end of the web scraping step a function to delete any proper name contained in the results that will later be sent to EchoBT. However, this considerably increases the execution time of the program, which is multiplied by at least 5. To save time while ensuring consistency of results, it would be possible to perform this step after receiving the results from EchoBT, but then some terms would be deleted without being replaced by others. Currently, no such processing is applied at the tool level.

- the proposed terms do not contain any accent because the algorithms Echo cannot take them into account. It would therefore be good to ensure that this is the case.

Finally, the execution time of the process (from the moment the user clicks on the button to launch the study until the results are displayed) should be optimized. Currently, this one lasts on average one minute but can be very variable depending on the request given: it is actually the web scraping step that takes the longest time to complete and it turns out that Scrapy has difficulty extracting the data from some pages. So I set up a system that orders Scrapy to stop its action on a page if it lasts more than 5 seconds, but the slowing down due to this slight problem is still present.

8 Conclusion

During this internship, I was able to see for the first time how a company focused on IT worked and the working methods that applied to it. The project allowed me to improve my skills in this sector, both in terms of back-end and front-end: I used many languages and I took on new frameworks while taking an interest in new fields such as web scraping, machine learning or UX Design. I learned to implement an application from start to finish, something I had never done before. Finally, I discovered a work method that suited me well since I worked most of the time alone while interacting regularly with the outside world for advice, opinions or solutions.

All these parameters lead me to say that the last three months have been extremely enriching, both professionally through the discovery of different ways of working and through the completeness of the project carried out and personally through the acquisition of many new skills.

9 Resumes

9.1 French

Le stage effectué sur une période de trois mois était à la fois riche et complet. Il permettait de prendre en main de nombreuses technologies à travers des domaines variés allant du back-end au front-end en passant par du machine learning. Le but du projet proposé était de créer un service permettant à un utilisateur d'optimiser le référencement de sa page web pour une requête donnée. Cette optimisation était notamment permise grâce à des algorithmes d'apprentissage machine dont l'entreprise avait accès. Afin de pouvoir les utiliser, une de mes tâches était de leur fournir des données relatives au contenu d'un certain nombre de pages référencées pour cette requête. Une fois ces données transmises et grâce à un degré de pertinence qui leur était donné, les algorithmes étaient capables de différencier une page bien référencée d'une page mal référencée et donc de filtrer le contenu pour renvoyer les termes à intégrer dans la page de l'utilisateur afin de gagner des places dans les résultats de la recherche. Enfin, un affichage optimisé de ces termes accompagnés d'informations et fonctionnalités supplémentaires a été fait.

9.2 English

The internship over a three-month period was both rich and well-furnished. It made it possible to take in hand many technologies through various fields such as the back-end, the front-end and the machine learning. The goal of the suggested project was to create a service allowing a user to optimize the SEO of his web page for a given request. This optimization was made possible by machine learning algorithms that the company had access to. In order to be able to use them, one of my tasks was to provide them data related to the content of a certain number of pages indexed for this request. Once this data was transmitted and thanks to a degree of relevance given, the algorithms were able to differentiate a well indexed page from a poorly indexed one and thus filter the content to return the terms to be integrated in the user's page in order to gain positions in the search results. Finally, an optimized display of these terms with additional information and features has been made.

Appendix

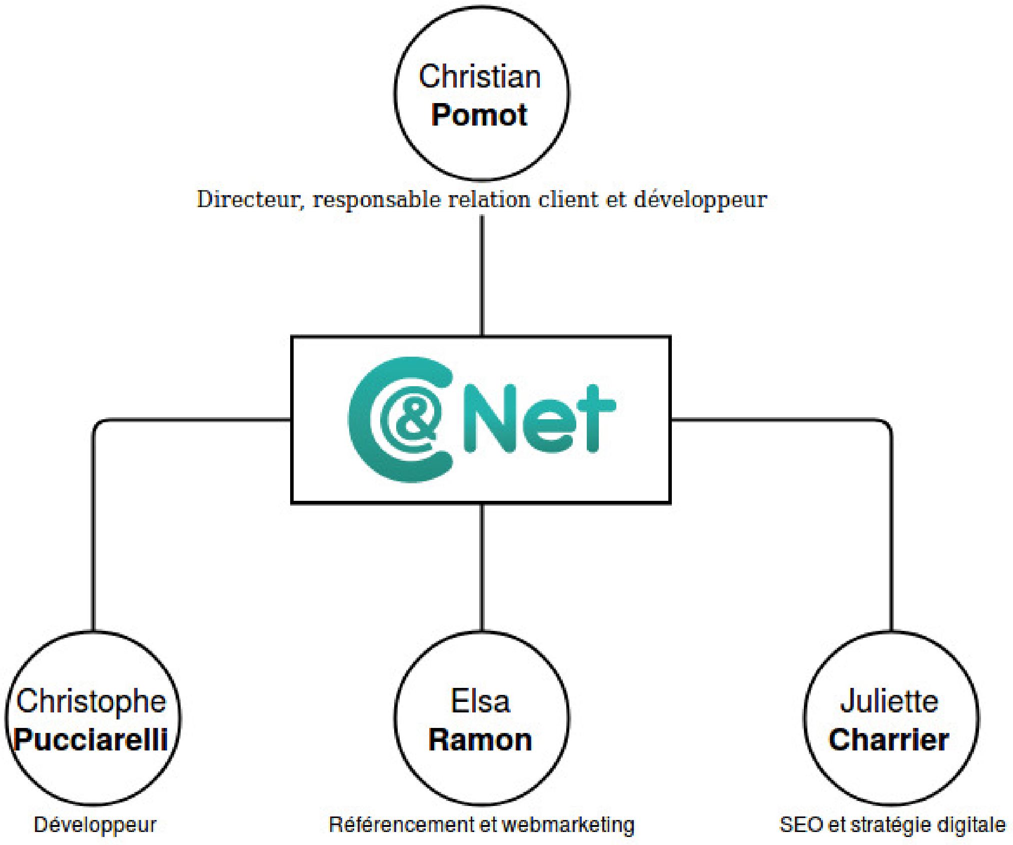
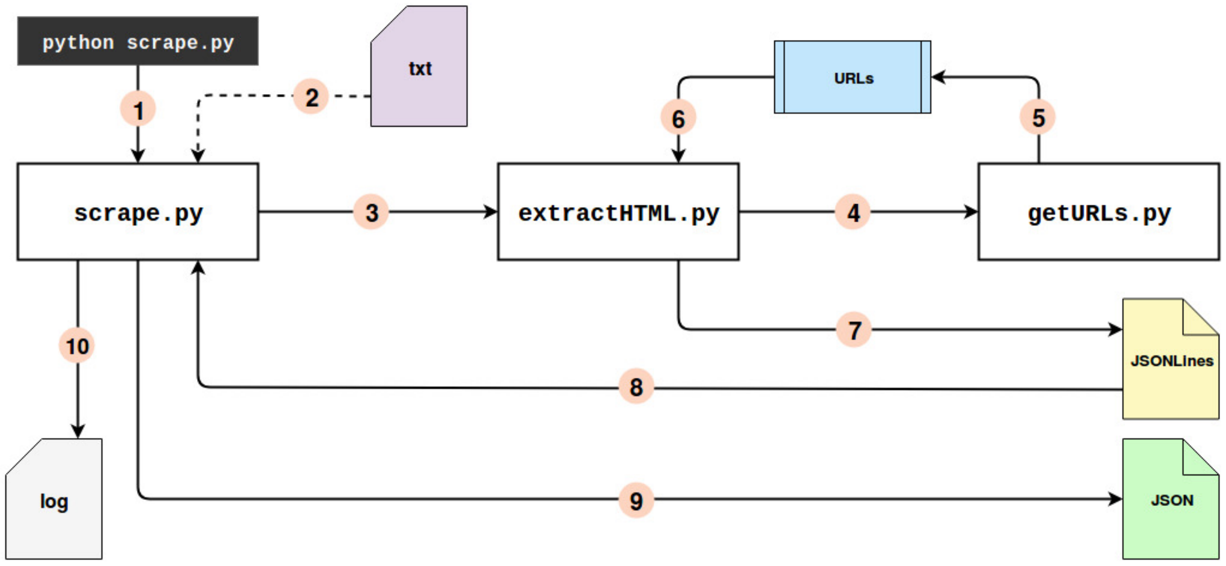

Christian
Pomot
Directeur, responsable relation client et développeur
C&Net
Christophe
Pucciarelli
Développeur
Elsa
Ramon
Référencement et webmarketing
Juliette
Charrier
SEO et stratégie digitale

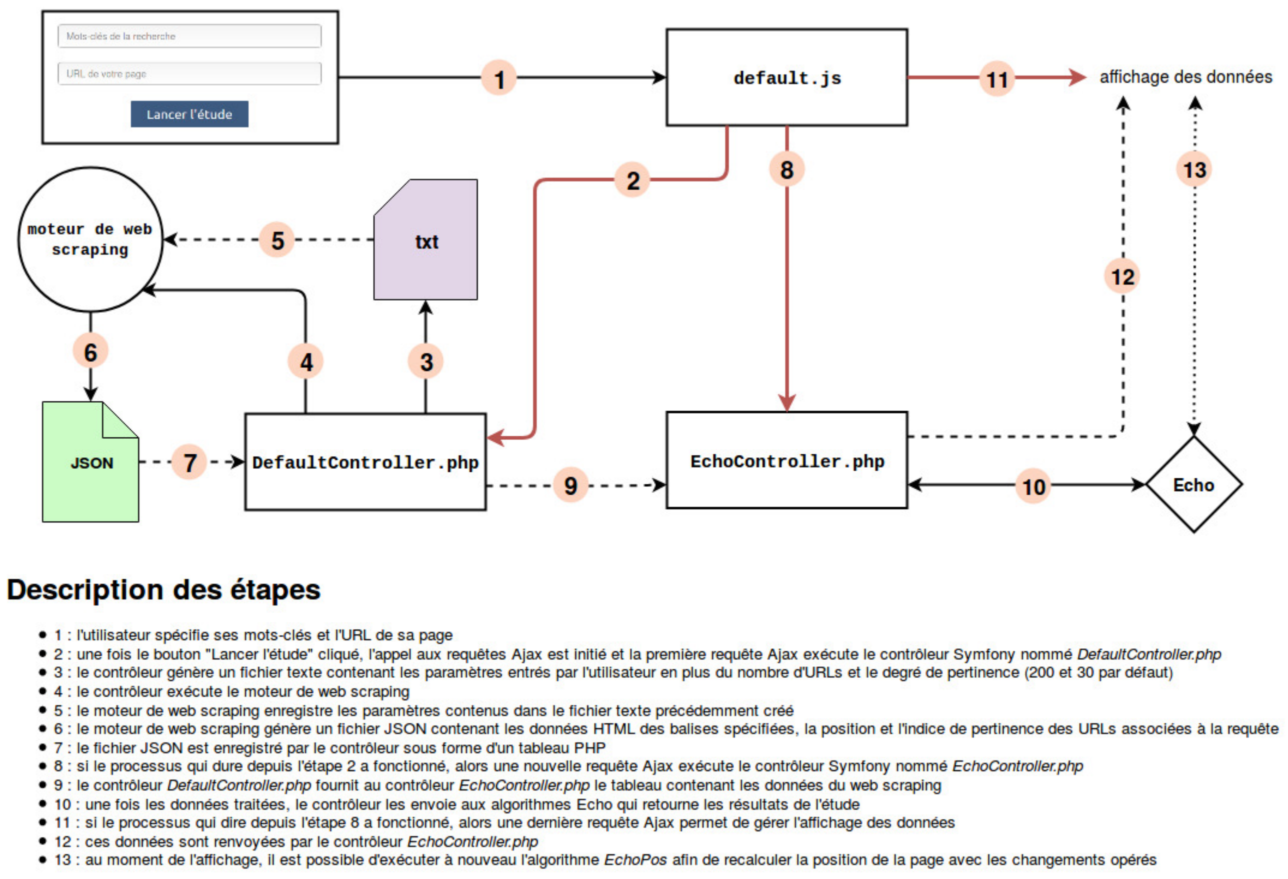
Figure 1: organization of the Com&Net company



Description des étapes

- 1 : le script *scrape.py* est exécuté via la commande tapée dans le shell
- 2 : les paramètres contenus dans le fichier texte sont enregistrés (URL utilisateur, nombre de résultats, degré de pertinence et mots-clés)
- 3 : le script *scrape.py* exécute le script *extractHTML.py* avec les paramètres récupérés
- 4 : le script *extractHTML.py* exécute le script *getURLs.py* avec les paramètres récupérés
- 5 : ce dernier renvoie une liste des n (nombre de résultats) premières URLs s'affichant suite à la requête (composée des mots-clés) donnée
- 6 : la liste d'URLs est renvoyée au script *extractHTML.py*
- 7 : le script *extractHTML.py* génère un fichier JSONLines comportant les données HTML des balises spécifiées, un indice de pertinence et une position pour chacune des URLs
- 8 : le fichier JSONLines est lu par le script *scrape.py*
- 9 : le script *scrape.py* génère un fichier JSON où chaque groupe de données associé à une URL est trié par sa position
- 10 : le script *scrape.py* génère aussi un fichier de log comportant toutes les informations liées au déroulement du processus

Figure 2: schematic representation of the architecture of the scraping engine



Description des étapes

- 1 : l'utilisateur spécifie ses mots-clés et l'URL de sa page
- 2 : une fois le bouton "Lancer l'étude" cliqué, l'appel aux requêtes Ajax est initié et la première requête Ajax exécute le contrôleur Symfony nommé *DefaultController.php*
- 3 : le contrôleur génère un fichier texte contenant les paramètres entrés par l'utilisateur en plus du nombre d'URLs et le degré de pertinence (200 et 30 par défaut)
- 4 : le contrôleur exécute le moteur de web scraping
- 5 : le moteur de web scraping enregistre les paramètres contenus dans le fichier texte précédemment créé
- 6 : le moteur de web scraping génère un fichier JSON contenant les données HTML des balises spécifiées, la position et l'indice de pertinence des URLs associées à la requête
- 7 : le fichier JSON est enregistré par le contrôleur sous forme d'un tableau PHP
- 8 : si le processus qui dure depuis l'étape 2 a fonctionné, alors une nouvelle requête Ajax exécute le contrôleur Symfony nommé *EchoController.php*
- 9 : le contrôleur *DefaultController.php* fournit au contrôleur *EchoController.php* le tableau contenant les données du web scraping
- 10 : une fois les données traitées, le contrôleur les envoie aux algorithmes Echo qui retourne les résultats de l'étude
- 11 : si le processus qui dire depuis l'étape 8 a fonctionné, alors une dernière requête Ajax permet de gérer l'affichage des données
- 12 : ces données sont renvoyées par le contrôleur *EchoController.php*
- 13 : au moment de l'affichage, il est possible d'exécuter à nouveau l'algorithme *EchoPos* afin de recalculer la position de la page avec les changements opérés

Figure 3: schematic representation of the architecture of the Symfony project

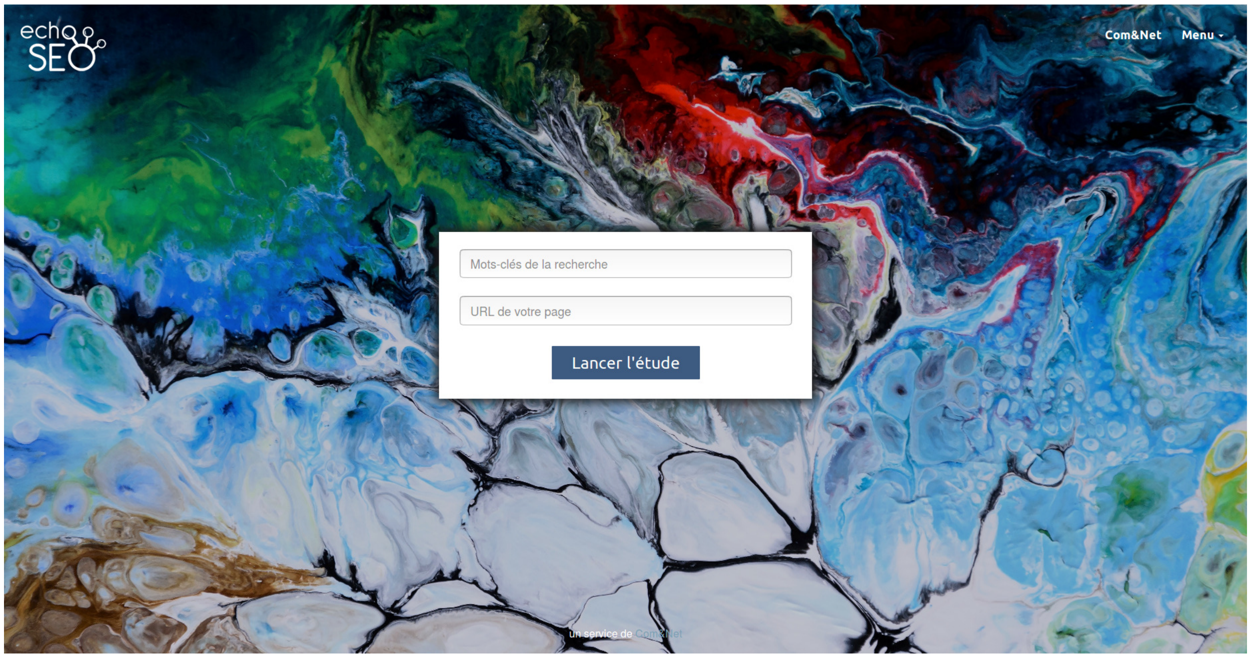
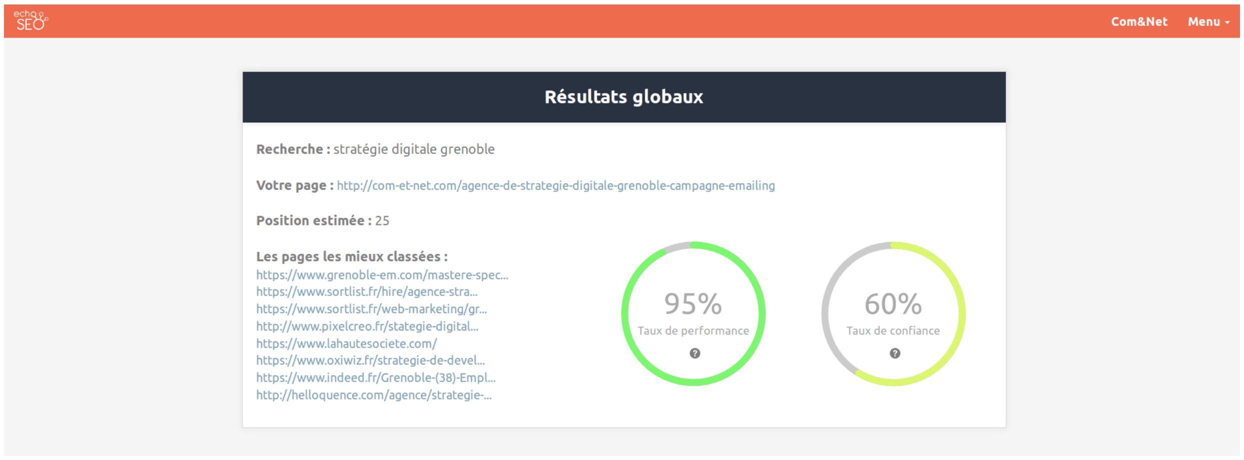


Figure 4: screenshot of the home screen



Résultats globaux

Recherche : stratégie digitale grenoble

Votre page : http://com-et-net.com/agence-de-strategie-digitale-grenoble-campagne-emailing

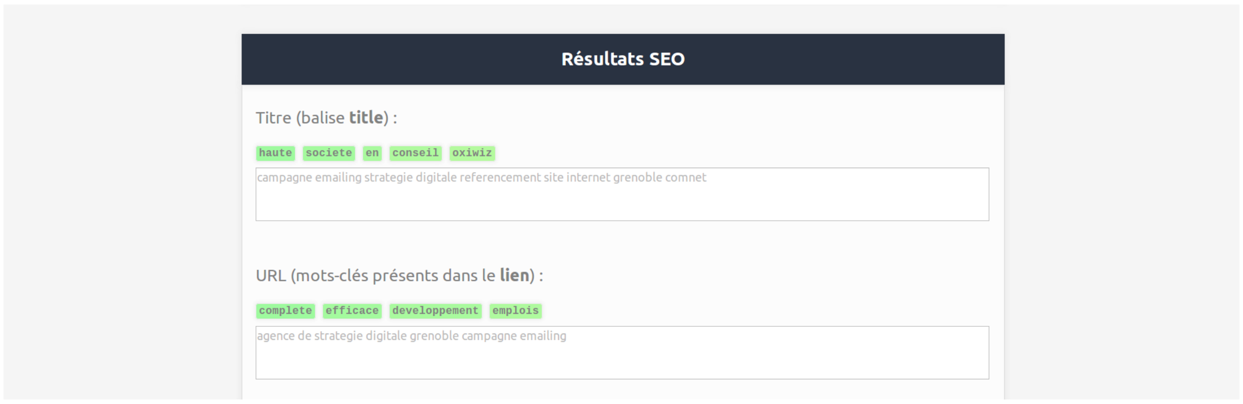
Position estimée : 25

Les pages les mieux classées :
https://www.grenoble-em.com/mastere-spec...
https://www.sortlist.fr/hire/agence-stra...
https://www.sortlist.fr/web-marketing/gr...
http://www.pixelcreo.fr/stategie-digital...
https://www.lahautesociete.com/
https://www.oxiwiz.fr/strategie-de-devel...
https://www.indeed.fr/Grenoble-(38)-Empl...
http://helloquence.com/agence/strategie-...

95%
Taux de performance

60%
Taux de confiance

Figure 5: screenshot of the top of the results page



Résultats SEO

Titre (balise **title**) :

haute societe en conseil oxiwiz

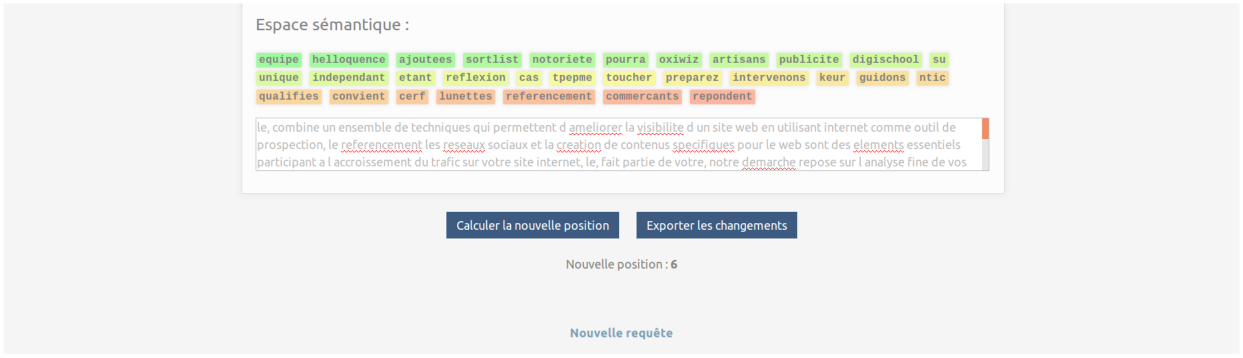
campagne emailing strategie digitale referencement site internet grenoble comnet

URL (mots-clés présents dans le **lien**) :

complete efficace developpement emplois

agence de strategie digitale grenoble campagne emailing

Figure 6: screenshot of the middle of the results page



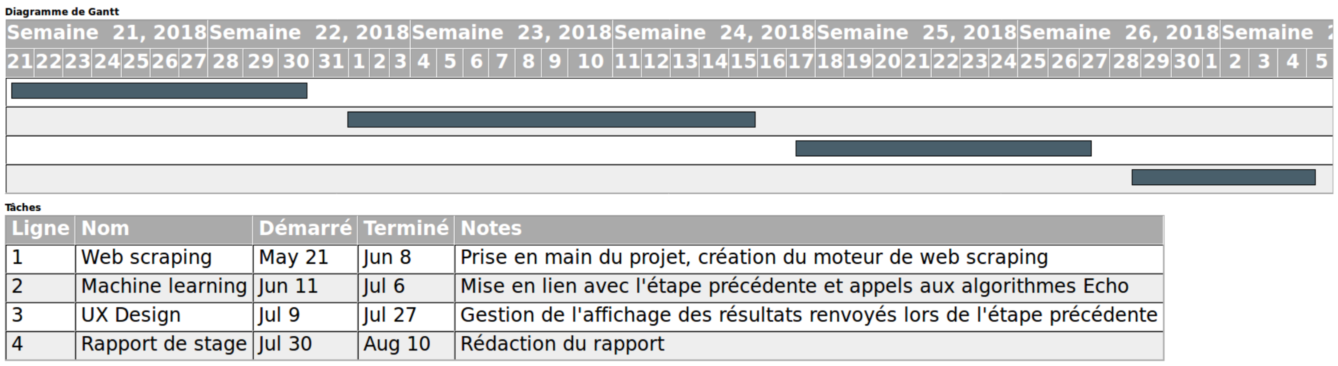
Espace sémantique :

equipe helloquence ajoutees sortlist notoriete pourra oxiwiz artisans publicite digischool su unique independant etant reflexion cas tpepme toucher preparez intervenons keur guidons ntic qualifies convient cerf lunettes referencement commercants repondent

le, combine un ensemble de techniques qui permettent d ameliorer la visibilite d un site web en utilisant internet comme outil de prospection, le referencement les reseaux sociaux et la creation de contenus specifiques pour le web sont des elements essentiels participant a l accroissement du trafic sur votre site internet, le, fait partie de votre, notre demarche repose sur l analyse fine de vos

Nouvelle position : **6**

Figure 7: screenshot of the bottom of the results page



Ligne	Nom	Démarré	Terminé	Notes
1	Web scraping	May 21	Jun 8	Prise en main du projet, création du moteur de web scraping
2	Machine learning	Jun 11	Jul 6	Mise en lien avec l'étape précédente et appels aux algorithmes Echo
3	UX Design	Jul 9	Jul 27	Gestion de l'affichage des résultats renvoyés lors de l'étape précédente
4	Rapport de stage	Jul 30	Aug 10	Rédaction du rapport

Figure 8: Gantt chart