Web IR Report

Louis Kusno
2025403026

Abstract

a

2

Contents

1 Background & Motivation

At first we wanted to make a search engine for humans. Much like OSINT where you can search for people and find their social media accounts, contact information, and other personal data. We thought this would be a great idea because it would allow people to find others easily and quickly.

However, we realized that most of the data available online is from social media sites such as: LinkedIn, Facebook, Twitter, Instagram, etc. And that scraping these sites is against their terms of service. So we decided to make a search engine for professors from the top ranked universities in the world. University web pages are publicly accessible and more "permissive" to crawl than LinkedIn.

What we ended up with is **Sorgle** (profes**sor** goo**gle**), a search engine that allows users to search for professors from the high ranked universities in the world.

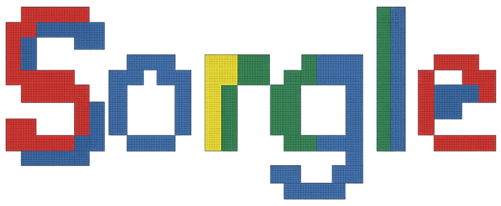


Figure 1: Sorgle Logo

2 Related Work

When the idea of the project hatched, we were thinking of OSINT (Open Source Intelligence) tools that allow users to search for people. This is a common practice in the field of cybersecurity, where professionals often need to gather information about individuals or organizations from publicly available sources. We found ContactOut [1], a website that essentially scrapes LinkedIn profiles and provides contact information for individuals.

Another notable OSINT tool is Mr.Holmes [2], which is designed to automate the process of gathering intelligence from various online sources. Mr.Holmes aggregates data from social media platforms, public records, and other web resources to help users build comprehensive profiles on individuals or organizations. Its modular architecture allows for the integration of new data sources, making it a flexible and extensible solution for OSINT investigations. The tool is widely used by cybersecurity professionals and investigators for its efficiency and breadth of coverage.

However, a big part of Mr.Holmes is the use of google dorking, which is a technique that uses advanced search operators to find specific information on the web. For example, a typical dork query to find publicly available PDF files on a specific domain might look like:

```
site:example.com filetype:pdf
```

This technique uses an already existing search engine, and it wouldn't make sense to build a new search engine that relies on it.

3 System Design Overview

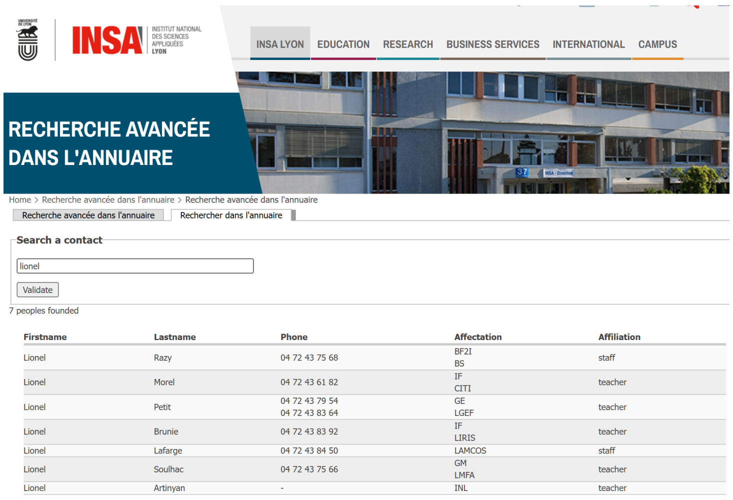
A search engine is a system that allows users to search for information on the internet. It consists of two main components: a crawler and a search engine. The crawler is responsible for collecting data from the internet, while the search engine is responsible for searching and displaying that data.

Since we are a team of two, we decided to split the work into two parts: one person will work on the crawler and the other person will work on the search engine. This way, we can work in parallel and complete the project faster.

3.1 Crawler

We both started by crawling our own universities, mine being INSA Lyon in France. I used **Selenium** to send requests and **BeautifulSoup** to parse the HTML content. **Selenium** was used because INSA Lyon's website is a single page application that only allows users to search people by name, as shown in Figure 2. This means that we cannot scrape the entire list of people from the website, but we can search for them by name. In the end, to get the full list of professors, I had to search through (with a rate limit to maintain politness) every two lettter combination of the alphabet (from "aa", "ab", "ac", ..., "zz") and store the results in a CSV file. Then clean it up by removing duplicates, students and staff.

Now that I had data from my university, I wanted to start designing the layout of the search engine. I wanted to have a simple and clean design that would allow users to easily search for professors



RECHERCHE AVANCÉE DANS L'ANNUAIRE

Home > Recherche avancée dans l'annuaire > Recherche avancée dans l'annuaire

Recherche avancée dans l'annuaire | Rechercher dans l'annuaire

Search a contact

lionel

Validate

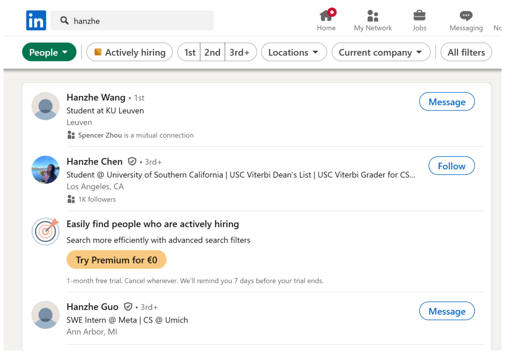
7 peoples founded

Firstname	Lastname	Phone	Affectation	Affiliation
Lionel	Razy	04 72 43 75 68	BF2I BS	staff
Lionel	Morel	04 72 43 61 82	IF CITI	teacher
Lionel	Petit	04 72 43 79 54 04 72 43 83 64	GE LGEF	teacher
Lionel	Brunie	04 72 43 83 92	IF LIRIS	teacher
Lionel	Lafarge	04 72 43 84 50	LAMCOS	staff
Lionel	Soulhac	04 72 43 75 66	GM LMFA	teacher
Lionel	Artinyan	-	INL	teacher

Figure 2: INSA Lyon's Results Page

3.2 Search Engine

I started by looking at different people search engines: LinkedIn, Instagram, Facebook, etc. I wanted to see how they display the results and what information they show. I also looked at different search engines such as Google to see how they display the results. Since the type of data we are dealing with is mostly related to the professors themselves, I wanted to display the name, title, and university of the professor as well as the picture if available. Therefore, I greatly inspired my design on LinkedIn's search results page (Figure 3 [3] and Figure 4) and its profile page (Figure 5 [3] and Figure 6).



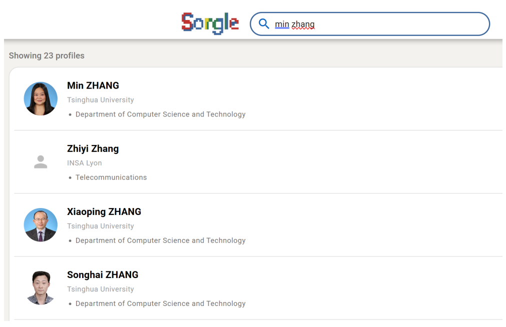
Hanzhe Wang • 1st
Student at KU Leuven
Leuven
Spencer Zhou is a mutual connection

Hanzhe Chen • 3rd+
Student @ University of Southern California | USC Viterbi Dean's List | USC Viterbi Grader for CS...
Los Angeles, CA
1K followers

Easily find people who are actively hiring
Search more efficiently with advanced search filters
1-month free trial. Cancel whenever. We'll remind you 7 days before your trial ends.

Hanzhe Guo • 3rd+
SWE Intern @ Meta | CS @ Umich
Ann Arbor, MI

Figure 3: LinkedIn Results Page [3]



Showing 23 profiles

Min ZHANG
Tsinghua University
• Department of Computer Science and Technology

Zhiyi Zhang
INSA Lyon
• Telecommunications

Xiaoping ZHANG
Tsinghua University
• Department of Computer Science and Technology

Songhai ZHANG
Tsinghua University
• Department of Computer Science and Technology

Figure 4: Sorgle Results Page

For the results page, I wanted to keep the design simple and clean, with a search bar at the top and the results displayed below it. I kept the information display very brief; the name, university and department because this

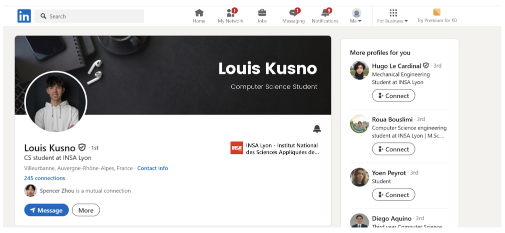


Figure 5: LinkedIn Profile Page [3]

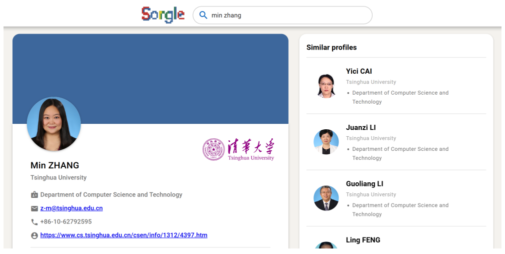

Sorgle
min zhang
Min ZHANG
Tsinghua University
清華大學
Tsinghua University
Department of Computer Science and Technology
z-m@tsinghua.edu.cn
+86-10-62792595
https://www.cs.tsinghua.edu.cn/csen/info/1312/4397.htm
Similar profiles
Yici CAI
Tsinghua University
Department of Computer Science and Technology
Juanzi LI
Tsinghua University
Department of Computer Science and Technology
Guoliang LI
Tsinghua University
Department of Computer Science and Technology
Ling FENG

Figure 6: Sorgle Profile Page

is the data that all profiles have in common. I also really liked LinkedIn's color scheme. Even thought I set all my apps to dark mode, I think that light mode allows for a better looking website.

For the profile page, I wanted to display more information about the professor, basically all the information that I could scrape from the university's website. This includes:

- Name
- University
- Department
- Profile picture
- Phone number
- Functions
- ORCID
- Bio

To display this information wihtout clutering the screen too much, I used logos that could best describe each field. Next, I really liked the idea of having similar profiles. This fills in the empty space on the right side of the page and allows users to use the search engine more efficiently (essentially increasing retention rate).

Lastly, the main page was inspired by Google's main page, this is coherent with my goal of having a simple and clean design.

Figure 7: Google Main Page [4]

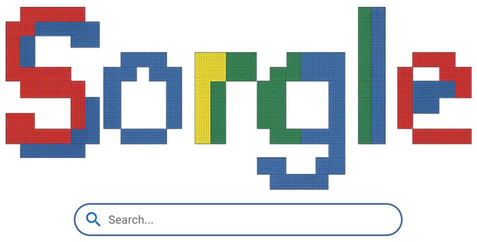


Figure 8: Sorgle Main Page

4 Technical Details

The faster and more relevant a search engine retrieves results, the better the user experience. In this section, we will discuss the technical details of our search algorithm.

4.1 Search Algorithm

The search algorithm is the core of the search engine. Instead of using basic string matching, we implemented a fuzzy search that allows for more flexibility. Fuzzy search is a technique that finds matches even when the search terms are not an exact match to the indexed data. This is particularly useful for names, which can have variations in spelling.

4.2 Fuzzy Search

The closeness of the search results is determined by the Levenshtein distance, which measures how many single-character edits (insertions, deletions, or substitutions) are required to change one word into another.

- insertion: cot $\rightarrow$ coat
- deletion: coat $\rightarrow$ cot
- substitution: cot $\rightarrow$ cat

4.3 Implementation

Fuzzy search was implemented on the name of the professors, their university and their department. Each with different weights. The name of the professor has the highest weight, followed by the university and then the department.

$$\text{score}(query, profile) \;=\; 0.6\, d_{\text{name}} \;+\; 0.25\, d_{\text{univ}} \;+\; 0.15\, d_{\text{dept}}\,,$$

This means that if a user searches for a professor's name, the search engine will prioritize results that match the name over those that match the university or department. I chose these weights based on the importance of each field as well as how long the field is. The name is the most important field but also the shortest, the deparemnt could be very long and therefore I wanted to give it a lower weight as it may skew the results.

The similar profiles are also determined by fuzzy search. This time the current profile's name, university and all department are used. The results are then sorted by the Levenshtein distance.

5 Experiments and Results

In this section, we will give an overview of the data we managed to scrape from the universities' websites.

Figure 9 is an overview of the whole data that we managed to scrape, this includes 4 universities: INSA Lyon, KU Leuven, Imperial College London, and Tsinghua University. And Figure 10 shows the most common first names among professors in the dataset.

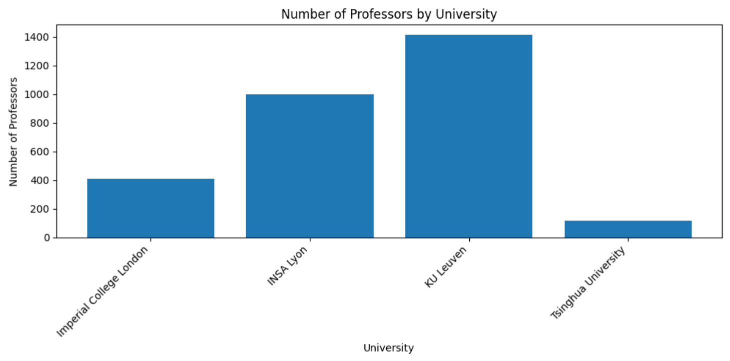


Figure 9: Number of Professors by University of the whole dataset

6 Conclusion & Future Work

We have presented a search engine that allows users to find professors based on their names, universities, and departments. However, there are several areas for improvement and future work.

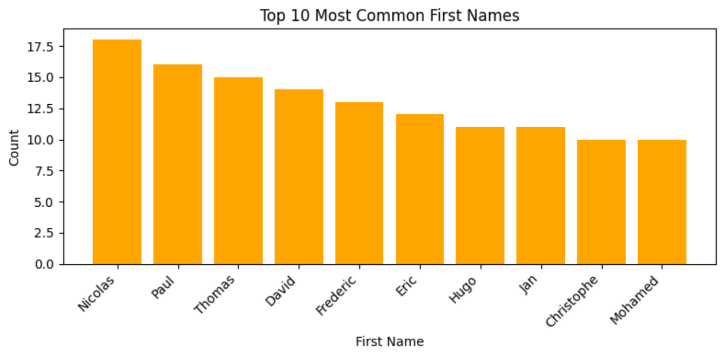


Figure 10: Most common first names among professors in the dataset

6.1 Future Work

The biggest limitation of our search engine is the lack of data. We only have a small subset of professors from four universities, which limits the search engine's effectiveness. To improve the search engine, we need to crawl more universities and gather more data. Not only the number of professors but also more fields such as research interests, publications, and full contact information would be beneficial. The easiest way to do this, would be to use the ORCID API, which provides a wealth of information about researchers, including their publications and affiliations.

Moreover, the search engine could be enhanced by implementing more advanced search features, such as filtering by research interests, publications or their role (eg: teacher, researcher, etc.). This would also have to be implemented for the similar profiles. However, I'm not sure if fuzzy search would be enough for the more advanced search features, setting up the weights for each field and the search algorithm would be a complex task.

Another simple but effective improvement would be to add an autocomplete feature to the search bar. Moreover, the current system stores all the data in a single json file. Creating a dedicated backend with a database would allow for more efficient data management and retrieval. This would allow us to implement profile views, top searched professors, and account management features.

Lastly, we have done minimal treatment to the data we crawled. One example is the names of the departments, which are often not standardized. For example, at INSA Lyon we have ”Computer Science”, at Tsinghua University we have ”Department of Computer Science and Technology”, and at Imperial College London we have ”Department of Computing”. Standardizing these names would improve the search engine's effectiveness as well as the user experience (especially for the similar profiles section).

6.2 Conclusion

In conclusion, we have built a simple search engine that allows users to find professors based on their names, universities, and departments. The search engine is functional and provides a good user experience, but there is still much work to be done to improve it. We hope that this project will serve as a foundation for future work in this area and that it will inspire others to build similar systems.

6.3 Ethical Use & Privacy.

All data we collect is publicly accessible. We do not scrape any password-protected or user-locked content. We also do not store or display student names—only faculty. Users of Sorgle must not use this information to spam or harass.

References

[1] ContactOut, "Contactout." https://contactout.com/. Accessed: 2024-06-10.

[2] Y. Wang and Y. Li, "Mr. Holmes." https://github.com/Lucksi/Mr.Holmes. GitHub repository, accessed June 7, 2025.

[3] LinkedIn Corporation, "LinkedIn: Professional Networking Platform." https://www.linkedin.com/, 2025. Accessed June 7, 2025.

[4] Google LLC, "Google Search." https://www.google.com/, 2025. Accessed June 7, 2025.