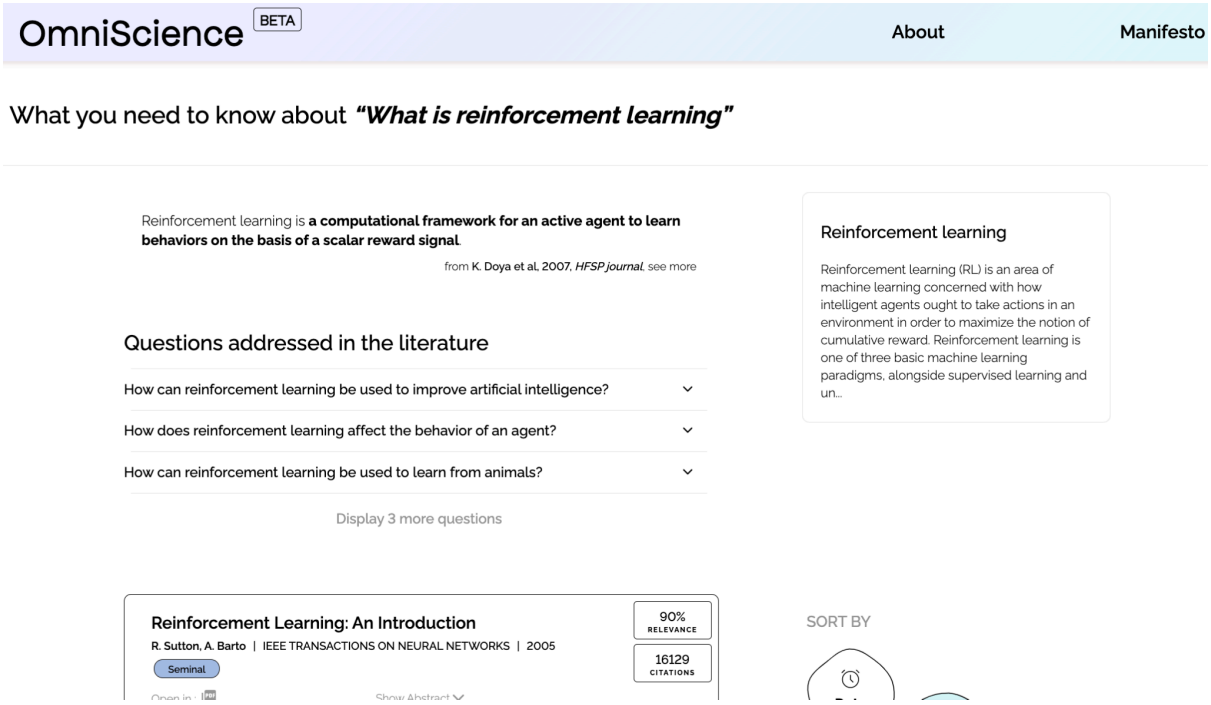
How we built a Google Scholar Engine on a 16Go RAM VM

Eight months ago, we decided to build a better Google Scholar. The reasons were the following:

- the information was not enough organized for our taste
- their engine didn't seem to have evolved since Methuselah
- For some reason, the quality of results on Google does not seem to be transferable to Google Scholar
- Microsoft Academic, the main serious competitor of Google Scholar, has announced its death for the end of December 2021.
- The results on Semantic Scholar don't seem to be crazy either
- They are not open source

We think we have succeeded in doing something.



We are proud to present [omniscience.academy](), which exploits both the structure of the graph of citations and topics as well as the semantics of the articles in order to propose results that are admittedly somewhat slow, but we believe also more accurate.

We rely on recent advances in NLP to extract the semantic of all scientific articles.

We admit Google Scholar is unbeatable on the recency of the results, but we A/B tested our algorithm against its results in Economics with researchers, who were surprised by the quality of the results.

If you do not need the very last articles of the last 2 months, our tool should suit you.

Our vision

We want to help researchers who are transitioning from one topic to another, or students who are beginning their research, to find the seminal articles of their discipline (labelled as "seminal" articles).

We would also like to help researchers to establish an overview of the different topics in a field:

- Questions: In a result page, we extract the main research questions, and for each question, a list of corresponding articles is selected.
- The taxonomy of topics: For each search, we propose a view with the taxonomy of the related disciplines. We propose to visualize the number of researchers who have worked on these issues in the past years, which allows to better distribute the research effort over the different disciplines and to promote more diversity in the conceptual space of possible research.
- The dashboard of the main questions by disciplines (wip): We believe that not all research questions are created equal. We would like to compile a priority ranking dataset by field of study and see if the research effort is well aligned with the effort required by the priority of the questions. For example, in medicine, for each disease, we would like to trace the research effort and see if there is proportionality between the importance of the disease and the research effort devoted to it.

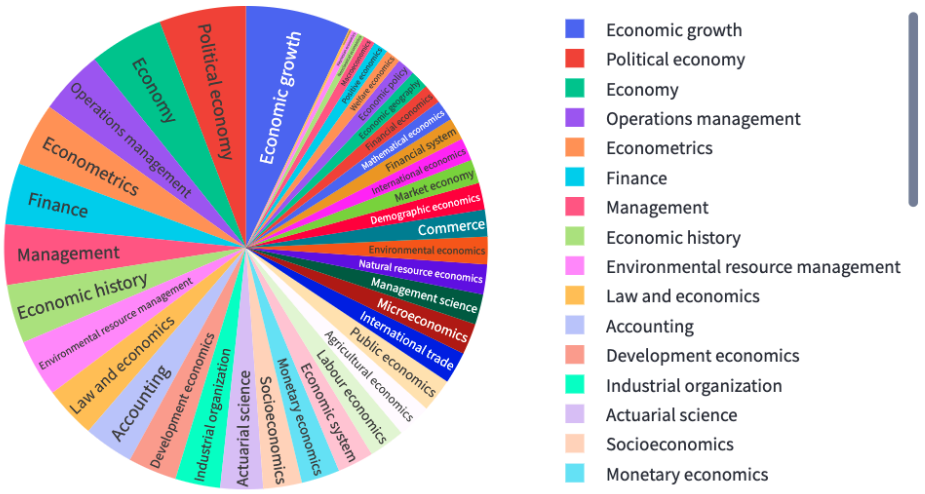
We hope that this segmentation work will make it easier for students to do their literature reviews and better allocate, at least locally, the brain resources.

Select one/multiple fields of study

Economics

Fields of study level 1

	displayname	papercount	fieldofstudyid
12	Economic growth	1012782	50522688
1	Political economy	837590	138921699
11	Economy	715043	136264566
17	Operations management	624999	21547014
22	Econometrics	612607	149782125
31	Finance	590394	10138342
3	Management	580785	187736073
2	Economic history	565673	6303427
0	Environmental resource m	553590	107826830
37	Law and economics	497813	190253527



Financial constraints

During the creation of our search engine, we realized the following paradox: even if academics and PhD students spend hours concocting their literature review, nobody really intends to pay for a search engine.

At the beginning of the project, we were only 4 students, so there was no question of paying too much money for the project: we booted the project with the means at hand.

With AWS, there are 10k dollars of credits available to startups for 2 years. The goal was to make a "better" Google Scholar for less than 10k dollars in about 10 months.

Oh, and if it wasn't already hard enough, at the beginning of the project, we didn't know how to do web development. Yeah…

Technical Resolution

It was hard, but it turns out it's possible. In three miracles:

The miracle of generating questions

Generating questions would have been impossible only two years ago, and is only due to the recent breakthroughs of NLP. But we have found a magic Byzantine recipe to create high quality questions without paying too much money with GPT3 for example.

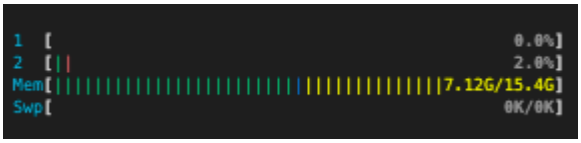
The miracle of our paper processing pipeline

In order to index the papers, we built our own pipeline.
We need to both summarize and find the main question associated with each paper. Then we have to index the embedding of the title, the abstract, and the question at the same time. Then we have to filter the bad entries with an expert Byzantine filter system.

Open source libraries already exist to index titles and questions like haystack, but we need an indexing speed much higher than what is possible on these libraries. Of course, Haystack has 4000 stars, and it's hard to do better than them in general. However, it is quite possible to do better when you focus on one sole metric: speed at the expense of feature versatility.

By focusing on indexing speed, we get a pipeline that indexes 100 times faster than the currently available open source tools.

The miracle of hosting a search engine on 8G of ram.



At the beginning of our journey, we used Elasticsearch to host our search engine. Elasticsearch is a wonderful technology, but it doesn't handle well the kind of scientific and technical words we want to get. Moreover, on AWS, the opensearch services are quite expensive and inaccurate on large scale data.

So we decided to host ourselves the search engine with a custom solution based on embeddings which allows semantic search, and keywords with Elasticsearch. And it turns out that it is possible to host both the title search engine, questions, and abstracts on an 8GB VM, (16GB to take a little margin). The main part of the trick is to memory map faiss-IVF indexes to an SSD, which is much cheaper than ram.

Of course, we have to fight quite a bit to optimize the recall, the compression, the query speed. We had to test about 20 different types of indexes before getting something satisfactory, but without the technical miracle of memory mapping we would not even have been able to host all the disciplines at the same time because of lack of resources.

Next Steps

If we get more users, we will set up precalculations that will theoretically allow us to get below 1 seconds per query with the current quality.

We will open source the project and the search engine calibration in a few weeks. [edit: Done! The code is open sourced here, you are welcome to contribute, to give advises or to criticize our algorithms. We especially welcome contributions to improve the front.]

We still have a lot to learn in terms of web infrastructure, if you have any advice on how to scale the search engine to hundreds of thousands of searches, we'll take it.

Future features:

- Even if we are happy with our algorithm, we still need to work on the underlying database of scientific articles, which is not yet complete (3 months)
- Better web interface to navigate between disciplines.
- Be able to find in one click the number of researchers working on each discipline/subdiscipline.
- Be able to follow live the progress on the main scientific issues of the moment.
- Let’s hope, joy and happiness