Projet Python

Puting order to the Scientific Literature

Cyril Verluise

May 30, 2018

Outline

Outline

Statement of purpose

The World Wide Web creates many new challenges for information retrieval. It is very large and heterogeneous (...) However unlike at document collections the World Wide Web is hypertext and provides considerable auxiliary information on top of the text of the web pages such as link structure (...) we take advantage of the link structure of the Web to produce a global "importance" ranking of every web page
L. Page, S. Brin, R. Motwani and T. Winograd (1999)

Statement of purpose

- Scientific literature is growing fast
- But: existing sorting and ranking methods are somehow disappointing.

  1. widely-spread scientific indexes such as the *h-index*, *i-index*, etc are mainly based on the number of citations without taking into account the field, or even the quality of the citing articles
  2. search engines such as Google Scholar are often disappointing and mostly work as a big articles' indexes

$\rightarrow$ mining the scientific literature is "more an art than a science"

Statement of purpose

There are risks:

- Harm the efficiency of scientists unintentionally engaged in research that already existed
- Harm research spillovers of research since non-scientists interested in research such as engineers, operation researchers, students, etc might be unable to find their way to the right existing knowledge

. . . and hope !

Like the World Wide Web, scientific literature is endowed with a directed link structure (citations) together with meta-information $\rightarrow$ attempt to use algorithms developed for the WWW to scientific articles and/or authors

Outline

Scraping Ideas-RepEc

- IDEAS: the largest bibliographic database dedicated to Economics and available freely on the Internet. It indexes over 2 500 000 items of research
- Based on RePEc a large volunteer effort to enhance the free dissemination of research in Economics which includes bibliographic metadata and citations from over 1 900 participating archives, including all the major publishers and research outlets
- Each article is accompanied with a set of useful features: title, author(s), date, journal, keywords, JEL, **AND** citations and references

Scraping Ideas-RepEc

Structure	Content
root	Homepage
+ article	List of editors' repositories
+ editor	List of journals' repositories
+ journal	List of articles
+ id	Article's pages

Table: Ideas-Repec architecture

Ex: `https://ideas.repec.org/a/oup/qjecon/v1y1886i1p1-27..html`

Focused on Top-30 journals (all time) articles $\rightarrow$ 82 000 articles. Each article's page was crawled and parsed to collect graph com- ponents (title, authors, citations and references) and attributes (date, keywords, JEL-Code)

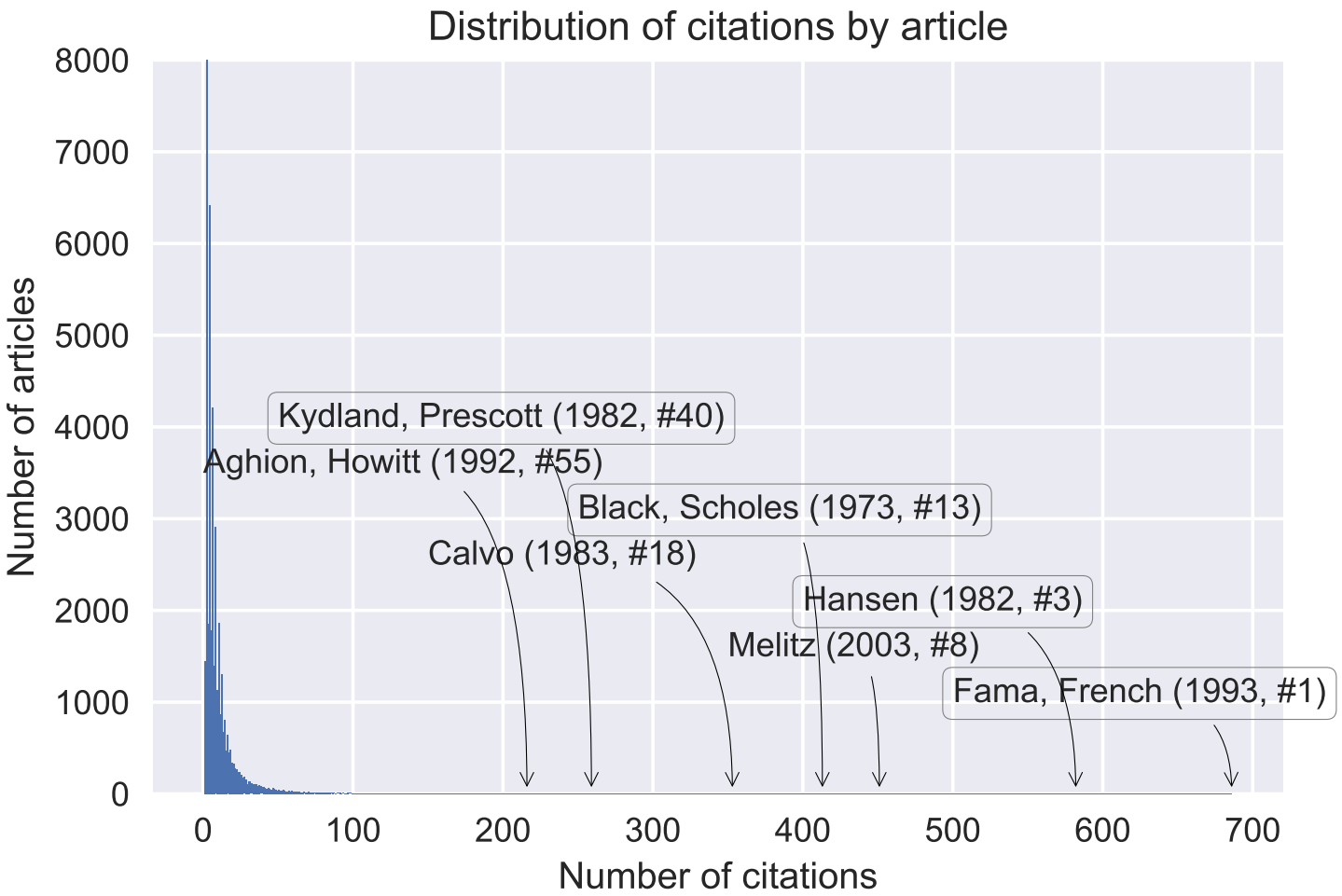
Name Disambiguation

- Issue: authors' name have different formats. Ex: Joseph Stiglitz appeared under no less that twenty aliases, sometimes including typos, such as J. Stiglitz, J. E. Stiglitz, Joseph Stiglitz ...
- Author names' disambiguation function, takes two authors as input and declares them similar or not based on how likely they are to be the same
  1. Levensthein distance with a threshold to match last names
  2. match the first names and the middle names the same way[1]

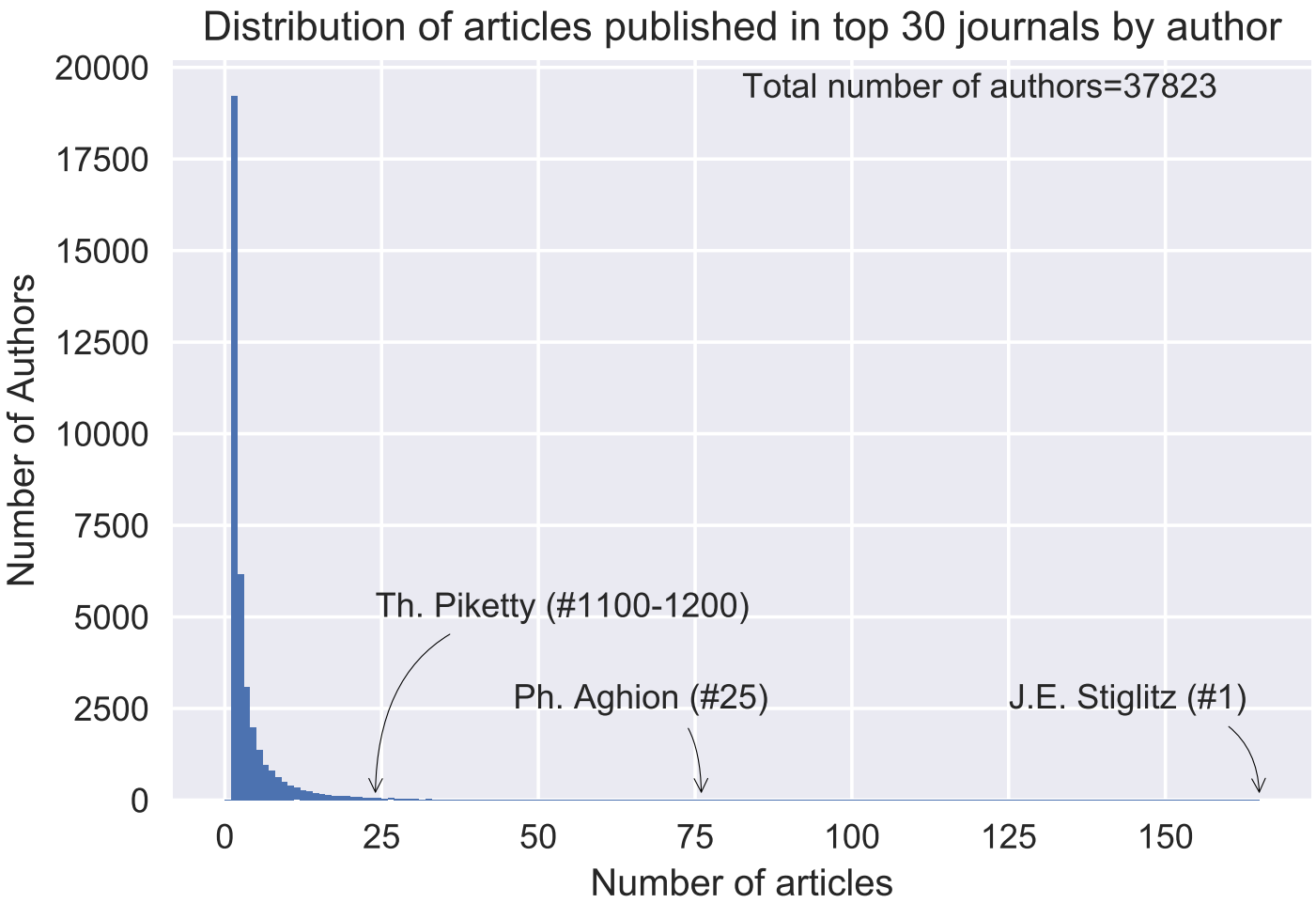
$\rightarrow$ from 52 000 to 37 000 different names

[1] special treatment of cases when the first name and/or the middle name are only one character or either one does not exist ...

Data validation

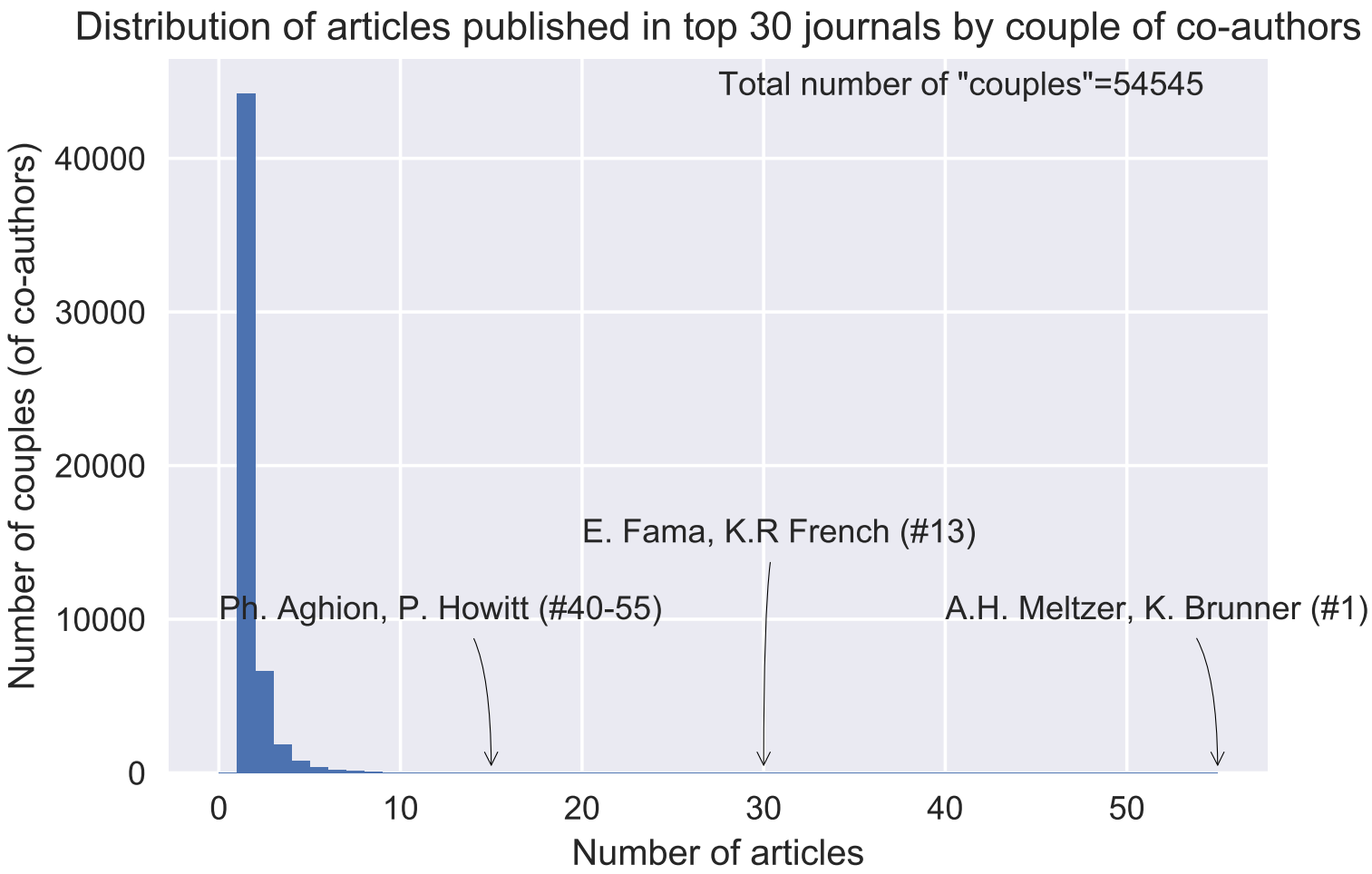


Data validation



Nb: 50% of authors that published at least one article in the top-30 journals published only once in these journals

Data validation



Some words on Rankings

Two "competing" algorithms

- PageRank (Page and Brin, 1998): ranking ($\approx$ random surfer) on whole graph + query filtering/rearranging
  - Graph is potentially huge and G mat is not sparse[2]
  - But matrix*vector is easy to distribute
  - Once and for all
- HITS (Kleinberg, 1999): query $\rightarrow$ iterative subgraph building $\rightarrow$ Hubs and Authorithies
  - Smaller graph
  - Sparse matrix
  - One query, One full computation process

[2] But can be written efficiently

Implementation

3 different rankings[3]

- Number of citations (benchmark)
- PageRank
- Pagerank with time discount[4]
- HITS[5]

[3] $\approx$ 50 000 articles have at least 1 citation, ie are ranked
[4] Accounts for scientific citations specific time structure
[5] full graph

Searching

2 approaches:

1. "Word filtering" on global ranking
2. Query specific ranking
   - Natural implementation of HITS (topic and article's similarity)
   - Personalized PaegRank (article's similarity)

Outline

Avenues for further work

- **Search engine !** (tf-idf, wod2vec, . . .)
- App deployment (Django, Flask, . . .)
- Extend dataset (arXiv, read pdf, . . .)
- . . .

External use

- Database could prove useful for many other research projects
- CitNet module as well[6]

[6] We provide full documentation and standard installation tools in this goal

Outline

Wrap-up

Technology used:

- Scraping
- Disambiguation
- Ranking + Network analysis
- Search
- overall Python embedding