Using Elasticsearch for entity recognition in affiliation disambiguation

Anne L'Hôte[1] and Eric Jeangirard[1]

[1]French Ministry of Higher Education, Research and Innovation, Paris, France

October 2021

Keywords: Elasticsearch, affiliation disambiguation, entity recognition

1. Introduction

The precise identification of the affiliations found in the bibliographic databases is a crucial point in various aspects and in particular to follow the production of one or a group of laboratories or institutions, and thus be able to observe trends related to publications at the institutional level.

Unfortunately, this exercise remains a complex task, giving part of the value of commercial bibliographic databases. Nevertheless, (Donner, Rimmert, and Eck 2020) have shown that relying solely on commercial databases is insufficient for any use with policy implications and that a specific cleanup effort is needed.

Some techniques have been proposed, based on supervised or semi-supervised approaches with clustering (Cuxac, Lamirel, and Bonvallot 2013). However, there are few, if any, labeled databases with an open license. These difficulties have led us, for the French Open Science Monitoring (Jeangirard 2019), to build our own methodology to detect publications with French affiliations.

This document aims at detailing a new methodology for linking affiliations to entities listed in international or national registries, in particular:

- Country list
- GRID ("GRID, Global Research Identifier Database" 2021)
- RoR ("RoR, Research Organization Registry" 2021)
- RNSR in France ("RNSR, Répertoire National Des Structures de Recherche" 2021)

- Sirene in France ("Sirene, Système National d'identification et Du Répertoire Des Entreprises et de Leurs établissements" 2021)

We propose a new approach, using the Elasticsearch search engine, based only on open data. This approach is built to be modular and easily adaptable to other international or local registries.

2. Method

2.1 Our matching framework

The problem we are looking for can be summarized as follows: let q be a string, describing an affiliation, and let be a set C { (condition_i, value_i) } (potentially empty) of additional conditions, giving structured information about the affiliation described by query. To fix the ideas, we can sometimes have extra informations, like the country of the affiliation, or the name of a supervisor. These informations should help the matcher to narrow down the possibilities. For example, in the case in which we know in advance the affiliation is in France, the set C will contain an element like $(country, France)$.

On the other hand, let R be a registry of entities (laboratories, institutions, even countries, cities etc). R is a set of objects with characteristics, such as, for example, in the case of a laboratory, one, or more name, acronym, address, supervisor, etc. The problem of affiliation recognition is to find the (potentially empty) set of elements of R that correspond to the query q and the conditions C.

Let's give an example. With q="French Ministry of Higher Education, Research and Innovation, Paris, France", and no condition set, using the the GRID repository, the expected result is https://grid.ac/institutes/grid.425729.f.

This task seems relatively simple to the human mind, but it is actually not so simple to automate. Rather than using a black-box technique, we propose a simple and modular approach where the user of the algorithm can keep control over the risk of error.

There are two types of errors in reality, precision (the proportion of false positives, i.e. how many times the algorithm gives a result that is a wrong match), and recall (the proportion of false negatives, i.e. how many times the algorithm does not give a match when a good one does exist).

Let us now introduce two concepts: **criterion** and **strategy**.

A **criterion** is a metadata describing the elements of the registry R (basically a field in the database). For example the entity name or the city is a criterion for the GRID registry. That is to say that this repository contains information about the names and cities of the entities that are in the registry. To take the previous example, the entity grid.425729.f has the following grid_country, grid_name and grid_city criteria (values of these criteria in the GRID registry):

- grid_city :
  - Paris
- grid_country :
  - France
- grid_name :
  - Ministry of Higher Education and Research
  - Ministère de l'Enseignement Supérieur et de la Recherche
  - Ministeri d'Educació Superior i Recerca francès

A **strategy** is a combination of one or multiple criterion. Thus, applying the strategy ['grid_city', 'grid_country', 'grid_name'], consists in returning all the elements of the registry R for which there is, at the same time, a match on the name, on the city and on the country with respect to the query received in input q and C. Using the same example, a single match is appropriate, giving the expected result:

- 'grid_city': ['Ministry of Higher Education, Research and Innovation, **Paris**, France']
- 'grid_country': ['Higher Education, Research and Innovation, Paris, **France**']
- 'grid_name': ['French **Ministry** of **Higher Education**, **Research** and Innovation, Paris, France']

2.2 Criteria and strategies

2.2.1 Direct criteria

Depending on the registry R and the nature of the registered objects, many criteria are possible. For example, a country repository could contain criteria like:

- the official name and the usual name of the country in different languages and their possible abbreviations (iso 3166 alpha-2, iso 3166 alpha-3)
- its subdivisions (regions, provinces…)
- its cities
- its street names, etc
- its institutions, universities, hospitals, etc
- its rivers, mountains, etc
- its Point of Interests

All of these are direct criteria, meaning they are direct characteristics that can be found as such in an affiliation text. For a research unit, think about an affiliation like "Institut des Géosciences de l'Environnement CNRS Saint Martin d'Hères". That string is actually a concatenation of:

- the research unit name: ""Institut des Géosciences de l'Environnement"
- the acronym of one the supervisors of the unit : "CNRS"
- the city of the unit : "Saint Martin d'Hères"

As explained above, the strategy combining the 3 criteria: name, supervisor acronym and city will match the correct entry in the RNSR registry.

2.2.2 Indirect criteria

In some cases, direct criteria such as these may be insufficient. Think about the example above, but with a slight modification : "Institut des Géosciences de l'Environnement CNRS Grenoble". Saint Martin d'Hères and Grenoble are two neighboring municipalities. Grenoble being much larger is sometimes used in the affiliation signature, even though the official address of the unit is with Saint Martin d'Hères. In that case, the above strategy combining name, supervisor acronym and city will give no match. A workaround is to use an indirect criterion, based on geographic proximity or influence. That could be a criteria like 'all cities within a radius of x kilometers'. Better, for France, INSEE has developed in the COG (French Official Geographic Code) ("Code Officiel Géographique (COG)" 2021) with several divisions, such as urban unit or employment zones (an employment zone is a geographic area within which most people reside and work). In itself, an employment zone has an identifier, but it is actually a list of all the cities that belong to the employment zone. That way, we can affect a criterion with the employment zone identifier to all the cities that belong to this employment zone. The employment zone identifier itself will generally not appear in the affiliation description, but still can be used as an (indirect) criterion.

If we take back the example above, during indexing, "Saint Martin d'Hères" will be catched as being part of employment *zone 8409*. At search time, with the query "Institut des Géosciences de l'Environnement CNRS Grenoble", "Grenoble" will also be catched with the criteria employment *zone 8409*, and so the strategy combining name, supervisor acronym and employment zone will return the correct match from the registry.

2.2.3 Strategies

The possible strategies are simply all the combinations of the criteria. A risk level can be associated to each strategy, depending on the risk of false positives. A strategy combining many criteria will give a high precision but a low recall.

Thus, in the case of a country matcher, the strategy composed of the only criterion 'name of the country' can be risky. For example, for an affiliation like "Hotel Dieu de France, Beirut, Lebanon", this strategy would propose two matchings: France and Lebanon. In this case, France is a false positive. A more demanding strategy, searching for both the country name and a city, can avoid this pitfall in this case.

We therefore propose to test several strategies, more or less demanding, starting with the safest (in terms of risk of false positives). That way, the user of the matching algorithm keeps the control on the risks it accepts, choosing himself

the balance between precision and recall through the choice of the strategies that can be used.

2.2.4 Filtering sub-matching results

Some fine-tuning can be applied to the results. The main one we developed is a post-filtering to remove sub-matching results. Let us give an example to show what a sub-matching result is. Think about an input affiliation description like "Columbia University Medical Center, New York, USA". The strategy combining name, city and country for the GRID registry naturally matches two entries :

- grid.239585.0 (Columbia University Medical Center)
  - name: **Columbia University Medical Center**, New York, USA
  - city: Columbia University Medical Center, **New York**, USA
  - country: Columbia University Medical Center, New York, **USA**

- grid.21729.3f (Columbia University)
  - name: **Columbia University** Medical Center, New York, USA
  - city: Columbia University Medical Center, **New York**, USA
  - country: Columbia University Medical Center, New York, **USA**

For each criterion, the second result (grid.21729.3f) has either the same match as the first result or a match that is contained in the first result match for the same criteria. Namely, "Columbia University" is contained in "Columbia University Medical Center" for the name criterion, and the other matches ("New York" for the city criterion and "USA" for the country criterion are the same. The second result is then a sub-match compared to the first result, and can be filtered.

2.2.5 Strategy groups

As explained above, different strategies should be tested, from the most demanding (meaning with many matching criteria) to the least demanding (few matching criteria). If a strategy give one or more result, the loop over the strategies can be stopped. However, it does not always make sense to have a strict order between the strategies. That would have no impact if only one registry entry (at most) could be matched, but, in some cases, the affiliation signature in input should be matched with multiple registry entries. In that case, stopping at the first result of the most demanding strategy may impact the recall, especially if all the entries to be matched are not described with the same amount of details in the affiliation signature. So, instead of having an ordered list of strategies, we implemented an ordered list of strategy groups. A strategy group is itself a set of strategies, all the strategies within the group being tested for the matching.

2.3 Implementation with Elasticsearch

Elasticsearch is a very powerful and modular search engine technology. We used the 7.8.0 Elasticsearch version, with the analysis-icu plugin. The implementation of our method is done in two main steps:

- index construction: loading the criteria, each corresponding to an index in Elasticsearch
- search: the actual matching, by applying a list of strategy groups.

The choice of strategies to apply is made at the matching level and not at the loading level. The user of the matcher can therefore control himself his level of risk of false positives (or in other words his precision / recall balance).

The matching leverage on the diversity of features offered by Elasticsearch, in particular queries of type

- **match_phrase** (all terms, consecutively and in the same order) for short criteria where we want an exact match (like city names, or acronyms).
- **match** with a *minimum_should_match* parameter. For example, with a *minimum_should_match* at -20%, meaning that at most 20% (rounded down) of the terms can be missing, the order and the consecutive character not being taken into account : for longer criteria like the names of laboratories or supervisors.

2.3.1 Percolation in Elasticsearch

One feature of Elasticsearch that is critical for the implementation of our matching method is percolation. Usually, Elasticsearch allows to store documents in an index, and then to perform a query on this index. A typical example would be to perform a search query "InstitutionXYZ" against an indexed containing documents like "InstitutenXYZ Paris France". The searched term is actually contained in some of the indexed documents, that are then retrieved by Elasticsearch, with a computed score. However, in our use case, we face queries that look like "Hotel Dieu de France Beirut Lebanon". With the regular Elasticsearch behaviour, the document "Hotel Dieu de France" is matched, but the document "CHU Fort de France" is also retrieved, because all of them have in common the token 'France'. One could try to get around this issue using the Elasticsearch computed score, but it can quickly become cumbersome.

Actually, the desired behaviour is to reverse the role of the search query and the indexed document. Percolation allows us to do this, i.e. to store queries in an index, and then to search for a document in this index. With percolation, "Hotel Dieu de France Beirut Lebanon" will match "Hotel Dieu de France" but not "CHU Fort de France".

All other implementation details can be read directly in the open source code made available.

2.3.2 Elasticsearch index building

The first step of the matching algorithm is a one-shot process (to be done once), building all the indexes in Elasticsearch, each index corresponding to the criteria that will be used during the second step (the matching itself).

Each index is built using a datasource, namely:

- pycountry python module ("Pycountry" 2021) for country iso-codes, name and subdivisions
- GRID for country name and code, cities, and institutions names / acronyms
- RNSR for French research units names, acronyms, codes, supervisors and cities / employment zone

A document in a given index encapsulates two elements :

- 1. a matching query (using percolation) for the criterion (with the Elasticsearch **match** or **match_phrase**)
- 2. a list of matching entries from the source registry.

For example, in the matcher_grid_acronym index, the indexed document corresponding to the acronym "IUN" looks like the following:

```
{
  "_index": "matcher_grid_acronym",
  ...
  "_source": {
    "ids": [
      "grid.257418.d",
      "grid.489012.6"
    ],
    "query": {
      "match_phrase": {
        "content": {
          "query": "IUN",
          "analyzer": "acronym_analyzer",
        }
      }
    }
  }
}
```

In particular, we observe two entries in the GRID registry share the same acronym "IUN". Overall, the count of elements in each index is in the next table :

index	number of elements in index
country_alpha3	250
country_name	437
country_subdivision_code	58
country_subdivision_name	4737
grid_acronym	29347
grid_city	14398
grid_country	222
grid_country_code	220
grid_name	150529
rnsr_acronym	5876
rnsr_city	507
rnsr_code_number	39718
rnsr_country_code	20
rnsr_name	23504
rnsr_name_txt	23504
rnsr_supervisor_acronym	339
rnsr_supervisor_name	645
rnsr_urban_unit	1922
rnsr_year	227
rnsr_zone_emploi	22541

2.3.4 Elasticsearch matching

The matching implementation itself is rather straightforward and relies and the Elasticsearch **percolate** queries.

2.4 Evaluation

For a given repository R, we fix the strategies to apply, allowing us to set up an automatic matching. We will apply this method in the following section for 3 types of matcher: at the country level, for the GRID registry and for the French laboratory repository RNSR.

3. Results

3.1 Gold standard

In order to test our methodology and our strategies, we use a set of 4,705 data. The set of data and the affiliation of each data is collected from the Pubmed API. Each data has 5 attributes : the affiliation name itself and RNSR, siren, grid, country. The affiliation name is a string but it can contain multiple affiliations. The other attributes (rnsr, siren, grid and country) are manually collected. This dataset let us compute the precision and recall of each matcher by measuring the difference between the expected result and the computed one.

3.2 Current matchers results

For the country matcher, we used multiple strategies, combining:

- institution name (from GRID)
- institution acronym (from GRID)
- city name (from GRID)
- country name (from pycountry)
- country iso-code (from pycountry)
- country subdivisions (from pycountry)

For the RNSR matcher, the tested strategies combines the following criteria (all coming from RNSR):

- code number
- name
- acronym
- city
- employment zone (deducted from the city)
- supervisor name
- supervisor acronym

And for the GRID matcher, the criteria used are (all coming from GRID):

- name
- acronym
- country
- country code
- city

On the gold standard dataset we compiled, the results (precision and recall) are detailed for 3 matchers (country, RNSR and GRID) in the next table:

matcher	precision	recall
country	0.999	0.97
RNSR	0.987	0.757
GRID	0.773	0.678

4. Discussion and conclusion

4.1 Findings

The first take-away of this work is that Elasticsearch is a great tool not only for search (of course) but also for matching purposes. A lot of text treatment features are already in place (there are plenty of built-in tokenizers and analyzers) and the percolate queries fit well the matching use case. On top of that, the highlights features (that highlight the matching terms in the input context) are really useful.

Besides, it is also clear that the quality of the matching result really depend on the affiliation signature fed as input into the matcher: is there enough information ? but not too much noise ? That is why we thina matcher should be able to adapt to different situations and that we implemened a modular approach where the strategies and the underlying criteria can be chosen at the search time without having to re-initialze all the indexes.

The result we get for country matcher and the RNSR registry, on the corpus we tested are very promising.

4.2 Limitations and future research

The first limitation is that this matcher, and in particular the chosen criteria and strategies should be tested against more data. That would probably highlights new issues to solve. We also plan to develop more matchers for the international registry ROR and the French registry Siren.

Software and code availability

The source code is released under an MIT license in the GitHub repository https://github.com/dataesr/matcher.

Data availability

The gold standard dataset we have compiled is available here.

References

"Code Officiel Géographique (COG)." 2021. https://www.insee.fr/fr/information/2560452.

Cuxac, Pascal, Jean-Charles Lamirel, and Valerie Bonvallot. 2013. "Efficient Supervised and Semi-Supervised Approaches for Affiliations Disambiguation." *Scientometrics* 97 (1): 47–58. https://doi.org/10.1007/s11192-013-1025-5.

Donner, Paul, Christine Rimmert, and Nees Jan van Eck. 2020. "Comparing Institutional-Level Bibliometric Research Performance Indicator Values Based on Different Affiliation Disambiguation Systems." *Quantitative Science Studies* 1 (1): 150–70. https://doi.org/10.1162/qss_a_00013.

"GRID, Global Research Identifier Database." 2021. https://grid.ac/.

Jeangirard, Eric. 2019. "Monitoring Open Access at a National Level: French Case Study." In *ELPUB 2019 23d International Conference on Electronic Publishing*. OpenEdition Press. https://doi.org/10.4000/proceedings.elpub.2019.20.

"Pycountry." 2021. https://github.com/flyingcircusio/pycountry.

"RNSR, Répertoire National Des Structures de Recherche." 2021. https://appliweb.dgri.education.fr/rnsr/.

"RoR, Research Organization Registry." 2021. https://ror.org/.

"Sirene, Système National d'identification et Du Répertoire Des Entreprises et de Leurs établissements." 2021. https://www.sirene.fr/sirene/public/accueil.