From text to data inside bibliographic records. Entity recognition and entity linking of contributors and their roles from statements of responsibility

Thomas Zaragoza, Yann Nicolas, Aline Le Provost

▶ **To cite this version:**

Thomas Zaragoza, Yann Nicolas, Aline Le Provost. From text to data inside bibliographic records. Entity recognition and entity linking of contributors and their roles from statements of responsibility. IFLA WLIC 2022 Satellite Conference on Artificial Intelligence, Jul 2022, Galway, Ireland. hal-03701628

HAL Id: hal-03701628
https://hal.science/hal-03701628

Submitted on 22 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IFLA WLIC 2022 DUBLIN

Submitted on: 14/06/2022

Title of the Satellite Meeting: New Horizons in Artificial Intelligence in Libraries
Date: 21/06/2022 – 22/06/2022
Location: Galway, Ireland

From text to data inside bibliographic records. Entity recognition and entity linking of contributors and their roles from statements of responsibility

Thomas Zaragoza
Labo, ABES, Montpellier, France.
E-mail address: thomas.zararagozasafe@gmail.com

Yann Nicolas
Labo, ABES, Montpellier, France.
E-mail address: nicolas@abes.fr

Aline Le Provost
Labo, ABES, Montpellier, France.
E-mail address: le-provost@abes.fr

Copyright © 2022 by **Thomas Zaragoza, Yann Nicolas, Aline Le Provost**.
This work is made available under the terms of the Creative Commons Attribution 4.0 International License: http://creativecommons.org/licenses/by/4.0

Abstract:

Sudoc is the french higher education union catalogue. It is run by Abes. As any large database (15 million records), Sudoc has some quality issues that can negatively impact the user experience or the database maintenance efforts, e.g. the process towards a LRM compliant catalogue.

Quality issues are diverse: data can be inaccurate, ambiguous, miscategorized, redundant, inconsistent or missing. Sometimes, they are not really missing, they are hidden, lost in some text inside the bibliographic record itself. For instance, contributor names and roles are transcribed from the document to MARC descriptive fields (statement of responsibility). Most of them have a corresponding access point that contains the normalized name and a relator code (to express the role) - optionally the identifier of an authority record. But in Sudoc, many records have contributor mentions in descriptive fields that are not identified in access points. Moreover, many access points lack a relator code.

This paper will describe our efforts to extract structured information about contributors and their role from the statements of responsibility to automatically generate the following data in access points: last name, first name, relator code and optionally identifier to link to www.idref.fr, the french higher education authority file. The first step is a named entity recognition task implemented through a machine learning (ML) approach. For the recognition of names, a pre-existing generic model (from Spacy library) is employed and retrained with ad hoc data, annotated by librarians through a dedicated annotation tool (Prodigy). For roles, a model is generated from scratch. The second step is an entity linking task. The linking of contributor names is achieved with Qualinka, a logical rule based artificial intelligence framework (LE PROVOST, 2017 IFLA conference).The linking of roles is

currently still being debated with a preference for either an entity linking model or a classification model over a rule based approach.

This pipeline is for Abes a first experience in adopting machine learning and building a generic approach with the librarian in the loop.

Keywords: Statement of responsibility, cataloguing, access points, ner, text classification, machine learning

---

1 Introduction

Sudoc is the union catalogue of French academic and research libraries. Library catalogues are old, large and highly structured databases, created and maintained by professionals with a strong quality ethos. Recently the development of new information technology paradigms has stressed the importance of data quality: 1/ on a web of linked data, the pollution of good data by less good data is a permanent risk; 2/ predictive models and decision making algorithms require the best possible training data to optimise results and minimise the generation of additional erroneous data.

As in any large database, quality issues in Sudoc 15 million bibliographic records are diverse: data can be inaccurate, ambiguous, miscategorized, redundant, inconsistent or missing. Sometimes, they are not really missing, they are implicit, hidden or lost in some text inside the bibliographic record itself. This paper will focus on one important and interesting case.

Contributor names and roles are transcribed from the document to be recorded in MARC descriptive fields (statement of responsibility, which we will be referring to as SoR from now on). Most of them have a corresponding access point that contains the normalised name and a function relator code (to express the role) - optionally the identifier of an authority record. But in Sudoc, many records have contributor mentions in descriptive fields that are not identified in access points. Moreover, 300 000 existing person access points lack a relator code.

This paper describes our current effort to extract structured information (data) about contributors and their role from the SoRs (text). The objective is to automatically generate or correct access points containing the following data: last name, first name, function relator code and optionally identifier to link to www.idref.fr, the french higher education authority file.

Our effort can be decomposed into two main parts: 1/ creating and evaluating a Name Entity Recognition (NER) model to extract *Person* and *Role* entities from the SoRs; 2/ linking the extracted role to the UNIMARC controlled vocabulary of roles via text classification. As this project is still a work in progress, the last part will give an overview of further work to be done.

2 Incomplete bibliographic records: Missing access points.

Given a UNIMARC bibliographic record with the following statement of responsibility (SoR):

```
200 1 $aHawking$fStephen Finnigan, réalisation$gStephen Hawking,
Stephen Finnigan, Ben Bowie, scénario$gJoe Lovell ; Tina Lovell ;
Arthur Pelling [et al.] acteurs
```

B200$f and **B200$g** UNIMARC fields encode the SoR as found on the document (mainly the title frame as a source for a motion picture or the title page for a book etc.). The SoR transcribes, "records", the original **text** found on the document, with minimal structuration: a separate SoR per role/function. Hence, a unique SoR can mention more than one person:

```
$gStephen Hawking, Stephen Finnigan, Ben Bowie, scénario
```

The following UNIMARC fields deal with access, not description. This **highly structured data** comes from the intellectual analysis of the document by the cataloger, not the transcription of the title page:

```
700 1 $3241177782 $aFinnigan $bStephen $4300
701 1 $3028590295 $aHawking $bStephen $4690
701 1 $3241177286 $aLovell $bJoe $4005
701 1 $3241177421 $aLovell $bTina $4005
701 1 $3241177588 $aPelling $bArthur $4005
```

Each line is called an access point, and refers to a unique person.

In `701 1$3241177421$aLovell$bTina$4005`, the subfields respectively refer to the linked authority record[1], the last name, the first name and the UNIMARC role code[2] of the person relative to this document.

It is very easy for a person reading the previous SoRs to count the number of different persons mentioned (6) then to compare to the amount of access points (5) and come to the conclusion that this record is lacking access points. But this conclusion is not trivial to reach automatically.

In the process of encoding the 5 access points, the cataloguer extracted person names from the 3 SoR fields, but seems to have dismissed or forgotten one person (Ben Bowie). We want to rely on machines to extract the missing person, and to predict which precise role (function) Ben Bowie has.

300 000 author access points lack a relator code – lacunae that we will try to fill. Missing access points in the Sudoc database are, by construction, more difficult to find: this task would require to know how many different names are mentioned in the SoR. That is precisely one of the objectives of the project, which we can accomplish using a named entity recognition model.

1 https://abes.fr/es/reseaux-idref-orcid/le-reseau/

2 https://www.ifla.org/files/assets/uca/unimarc_updates/BIBLIOGRAPHIC/u_b_appb_update2020_online_final.pdf

3 ‘Person’ and ‘Role’ entity extraction (NER)

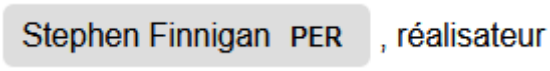
Named Entity Recognition (NER) is an application of Natural Language Processing (NLP) on large amounts of unstructured text to extract entities. We’ve decided to apply open-source models offered by spaCy[3] just for this purpose.

The model we use is ‘**fr_core_news_lg**’, that has been trained on 7200 high quality, hand annotated french articles on wikipedia (WIKINER) and over 3000 french sentences (French Sequoia). It can be used to extract persons, organisations and locations but not roles. For the latter, a new model is created and trained from scratch.

By applying the initial model on the statement of responsibility:

Stephen Finnigan, réalisateur

we obtain:



And whilst it may offer promising results in a broad context, this result doesn’t translate as much in a bibliographic context. Statements of responsibility are not written as naturally as the articles and sentences this model was trained on. Moreover, this model ignores the *Role* entity. So it was important to be able to retrain this model in a bibliographical context, and for this we had to annotate a large quantity of bibliographic records.

To do this, we have used Explosion’s[4] annotation tool: Prodigy. Prodigy is an annotation tool that allows us to annotate a greater amount of data faster and easier, with spaCy’s models pre-built-in for training and retraining models. While we can retain spaCy’s capacity for recognizing people, it has no capacity to recognize roles.

After annotating 10 000 bibliographical records using Prodigy, we were able to retrain spaCy’s model to better extract *Person* entities and begin extracting *Role* entities.

Evaluation

A new model can be evaluated using three metrics: the precision, the recall and the accuracy. These metrics can be visualised using a confusion matrix.

3 https://spacy.io
4 Team behind development of spaCy.

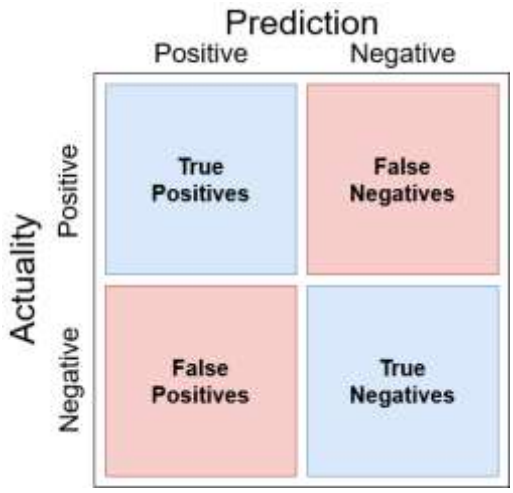


Figure: Confusion matrix example

The precision is the chance for a prediction to be true. It is calculated by dividing the amount of true positives by the sum of true positives and false positives. The recall is the chance for an entity of a certain class to be predicted as such. It is calculated by dividing the amount of true positives by the sum of true positives and false negatives. Finally, the overall accuracy of a model is the sum of true positives and true negatives divided by the sum of all four.

To return to our previous problem, after retraining our model, we have obtained the following results:

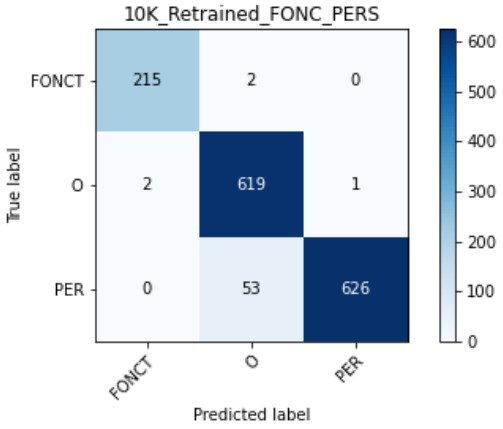


Figure: Confusion matrix of our spaCy model retrained on 10 000 records

	Precision	Recall	F1-Score	Support
Fonct (Role)	0,99	0,99	0,99	217
O[5] (Outside)	0,92	0,99	0,96	622
Per (Person)	0,99	0,92	0,96	679

We have a fairly performant model that is able to detect 99% of *Person* and *Role* ('FONCT') entities of which 99% of if its predictions are correct. This model is efficient enough for us to

5 'O' stands for outside of an entity. It is a token (word) that is neither *Role* nor *Person*.

tackle our next objective while still being able to further retrain our model, given additional annotated data.

To reiterate, we can now use a NER model to analyse statements of responsibility to detect the people mentioned and the keywords that represent their role in the creation of the catalogued document.

4 Linking role keywords to their role code using a text classification model

Now we want to link role keywords to their controlled role code. To illustrate the issue, let's once more rely on the following example:

Stephen Finnigan, réalisateur

There are around 155 different relator codes. For a human it is quite easy to link 'director' ('réalisateur') in the context of a movie to the function relator code *300 - Movie director*. But there can be difficulties clearing ambiguity without enough context. A director could also be an *632 - Artistic director*, *727 - Thesis Director* etc.

To automatically remove this ambiguity, we have first approached an Entity-Linking model. An entity-linking model consists in linking entities of a text to a controlled ontology. The problem we encountered with this approach was that it relied too much on a rules-based system, which could produce excellent results, but also required a larger quantity of work that was not viable within the context of this project.

So instead we have opted for a text classification approach. Text classification is a machine learning technique that assigns a set of predefined categories to open-ended text through various variables like word count frequencies and other predefined features. This would require no set of rules. The human work is limited to choosing features, extracting data, creating training sets, and choosing an algorithm that provides the best result for our problem.

Initially, we have decided to provide the following features: the role keywords[6], the document content type[7], the position of the mention of responsibility[8] and the document type[9] (If thesis or not).

Creation of the training dataset

To get the previous features and create a training set, we extract from the Sudoc database the entries for the UNIMARC fields 200$ (SoR), 70X$ (access points), 181$c (content type), 608$3 (ID of the authority record of the form/genre of the document), 105$a (textual resources types) and 503$a (Form title).

With our NER model we have extracted from the SoR the role keywords and the person names. From which we will pair these names with the persons in the access points to provide a relator code 'answer' for the training of our model.

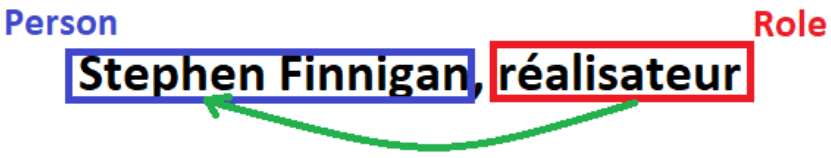
6 'Role' entities extracted by our NER model (examples: 'directed', 'written', 'illustrated', etc.)
7 One or more values from that list here. (examples: text, still image, animated image (video), etc.)
8 Position of a mention relative to other SoRs in the same record
9 We currently only exploit whether or not a document is a thesis / academic paper or not.

Here are the main steps of this process:

Once the raw data is extracted we note the position of each SoR, deduce whether the document is a thesis or not. We obtain the following entries:

Bibliographic Record ID	SoR	Position	Access Points	Content type	Thesis
236018256	Stephen Finnigan, director	0	Finnigan Stephen 300 Hawking Stephen 690 Lovell Tina 005 Lovell Joe 005 Pelling Arthur 005	tdi	False
236018256	Stephen Hawking, Stephen Finnigan, Ben Bowie, scénario;	1	Finnigan Stephen 300 Hawking Stephen 690 Lovell Tina 005 Lovell Joe 005 Pelling Arthur 005	tdi	False
236018256	Joe Lovell, Tina Lovell, Arthur Pelling [and al.] acteurs	2	Finnigan Stephen 300 Hawking Stephen 690 Lovell Tina 005 Lovell Joe 005 Pelling Arthur 005	tdi	False

The most complex step is executing our NER model on the SoRs to extract persons and roles keywords. The keywords are linked to the appropriate persons using a simple **Person*** + **Role*** / **Role*** + **Person*** pattern:



Afterwards we make a string comparison of the *Person* entity and the persons in the access points taking in consideration abbreviations, composite names and eventual typos using the Levenshtein distances between the two names. Given the NER model has correctly extracted the entities, the extracted name and access point name have been paired, we can create the following training set:

Keywords	Position	Content type	Thesis	Label
réalisateur	0	tdi	False	300
scénario	1	tdi	False	690
acteurs	2	tdi	False	5

Our approach has limits. Since it is vital that the training set is as accurate as possible we must ignore any ambiguity encountered that we are unable to remove. For example, if this bibliographic record was correctly filled out, Stephen Finnigan would have the following roles: 300 and 690. And in such case, it would be impossible for us to be able to distinguish between the associated keywords and function relator codes in the following entry:

Bibliographic Record ID	Keywords	Position	Content type	Thesis	Label
236018256	Directeur, scénario	0,1	tdi	False	300,69

‘Directeur’ could be associated with the role code 300 or 690 and it requires a decision to be made to choose between two… which is the objective of our model. A conundrum as we can not use the model’s predictions to create its own training set. At least not yet.

The inability to differentiate between multiple role codes is quite limiting and a root for discrimination for the training of our model as some roles have a higher chance of appearing in conjunction with others than by itself. So these roles tend to have less training entries.

Nonetheless we were able to create a training set of 1000 entries per function for 13 of the most common functions: *005 – Actor , 065 – Auctioneer, 070 – Author, 100 – Original Author, 230 – Composer, 300 – Movie director, 340 – Scientific publisher, 365 – Expert, 440 – Illustrator, 651 – Publication director, 727 – Thesis director* and *730 – Translator*. And created then trained a KNN algorithm model

We are limiting ourselves to 1000 entries as whilst we may have had hundreds of thousands of annotated training entries of certain classes (like the role code *070 – Author*) we only have thousands of entries of minority classes (like the role code *300 – Composer*). To avoid creating a biassed predictive model we have to create a balanced training set. The easiest and safest method in our case was ‘downscaling’, which entailed creating a training set with an equal amount of entries per role code. That amount is the highest possible while still remaining inferior to the amount of training entries from the lowest count class.

K-Nearest Neighbour (KNN)

“*If it looks like a duck, if it sounds like a duck, chances are… it’s a duck.*”

We first implemented the KNN model as an initial test to study the feasibility of our approach before branching out and trying various different algorithms and choosing the best suited, because the KNN algorithm is fairly simple to comprehend and makes for a nice introduction to the world of machine learning and decision making algorithms.

The entries of our training set are encoded into numerical values that can be computed by our model. Each training entry is then a point in a universe. For example, let’s say we train a KNN model with k=3 and two classes. Author is in red and Illustrator in blue.

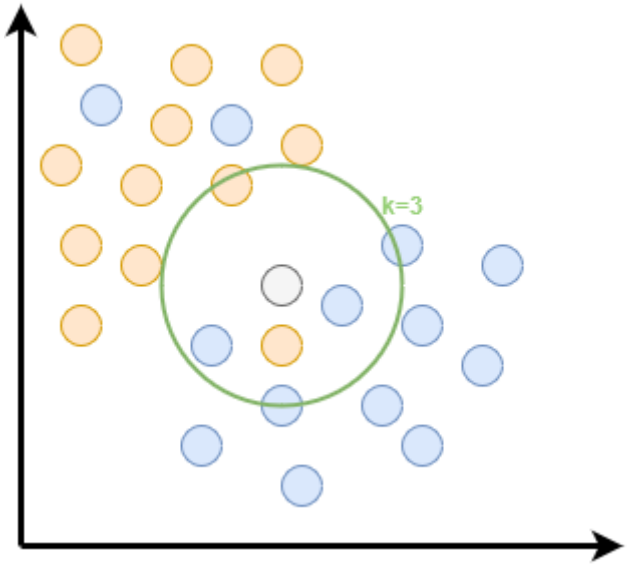


Figure: Prediction by KNN model with k=3

When attempting to predict the classification of a new point (grey in this case), we will look for its 3 nearest neighbours and predict the majority. In this case, the role *Illustrator* in blue.

Evaluation

After training our model, we can visualise the results in the following table and confusion matrix:

Label	Precision	Recall	F1-Score	Support
005	0,99	0,97	0,98	224
065	0,93	0,78	0,85	211
070	0,44	0,91	0,6	196
100	0,96	0,81	0,88	217
230	0,99	0,98	0,98	179
300	0,92	0,98	0,95	191
340	0,95	0,61	0,74	201
365	0,97	0,9	0,93	181
440	0,96	0,78	0,86	197
651	0,81	0,6	0,69	58
690	0,94	0,96	0,95	206
727	0,95	0,99	0,97	207
730	0,99	0,89	0,94	205

Figure: Confusion Matrix for KNN model, K=6 , 13k training

That shows an overall decent prediction model, given the limited amount of data used to train and test it.

The greatest discrepancy is in the *070 - Author* column of predictions. This class has a bad precision of 0.444 which is due to the nature of said class. Being the most common and principal role in the Sudoc, often in first position and not introduced by a role keyword, we implicitly arrive at the conclusion that said person is the author. Thus this lack of role keywords make this class a sort of 'collective bin' for all other entries that either lack role keywords (due to either a mistake by the NER model or initial cataloguing) or are simply too different to the average entry of its class. In practice since the class *070 – Author* is a majority role (see pie chart on distributions of roles per access point), the fact that class *070 – Author* is the de facto prediction for sparse entries is the least damaging to overall efficiency. And the decent recall and precision scores for other classes assure that the more specific and least common classes are correctly predicted.

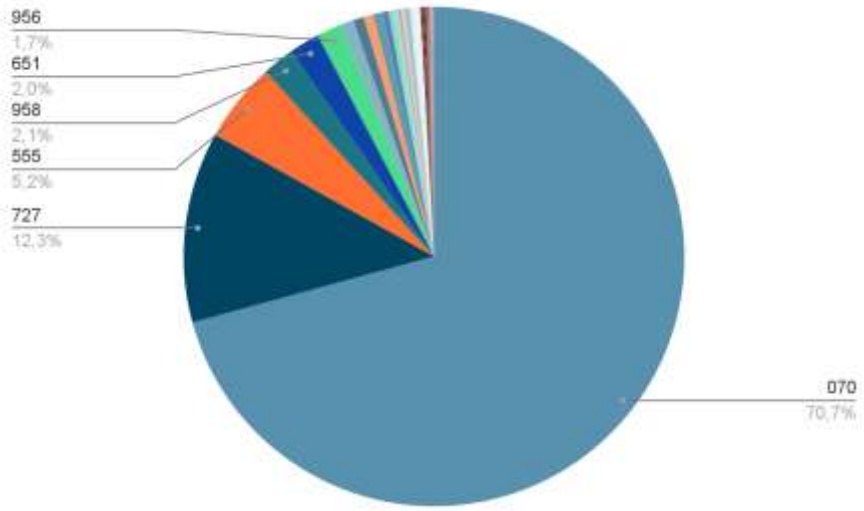


Figure: Pie chart of the distribution of function relator codes per access point for all records

As mentioned previously though, there are over 100 roles that our model needs to be able to differentiate and predict between. Considerably more than 13 roles. Thus it is unreasonable to believe our current approach is not flawed. Indeed if we train another model, this time differentiating between 35 classes, we obtain the following result:

Label	Precision	Recall	F1-Score	Support
003	0,67	0,01	0,03	114
005	0,9	0,99	0,94	192
010	0,84	0,69	0,76	146
040	0,15	0,24	0,18	169
065	0,81	0,78	0,79	185
070	0,37	0,9	0,53	194
080	0,84	0,81	0,83	199
100	0,74	0,74	0,74	196

180	0,96	0,93	0,95	222
205	0,5	0,78	0,61	218
212	0,82	0,8	0,81	175
220	0,76	0,59	0,66	136
230	0,91	0,94	0,93	203
273	0,47	0,44	0,45	213
300	0,92	0,95	0,94	198
340	0,67	0,35	0,46	204
350	0,62	0,86	0,72	196
365	0,96	0,88	0,92	199
410	0,93	0,87	0,9	190
440	0,83	0,62	0,71	213
460	0,76	0,64	0,69	181
470	0,71	0,37	0,49	147
550	0,92	0,88	0,9	132
555	0,38	0,08	0,14	173
595	0,53	0,25	0,34	201
600	0,97	0,72	0,83	199
651	0,61	0,47	0,53	185
673	0,21	0,86	0,34	217
690	0,97	0,86	0,91	222
710	0,88	0,59	0,7	192
727	0,85	0,08	0,16	198
730	0,99	0,88	0,93	210
956	0,88	0,51	0,65	192
958	0,82	0,48	0,61	205

We have outlined the cells with a score inferior to 0.7 and we can deduce this time that there are many more ‘collective bins’. This is due, in part, to the similarities between these roles that did not exist in the previous. If we chart out (simplified) the mispredictions:

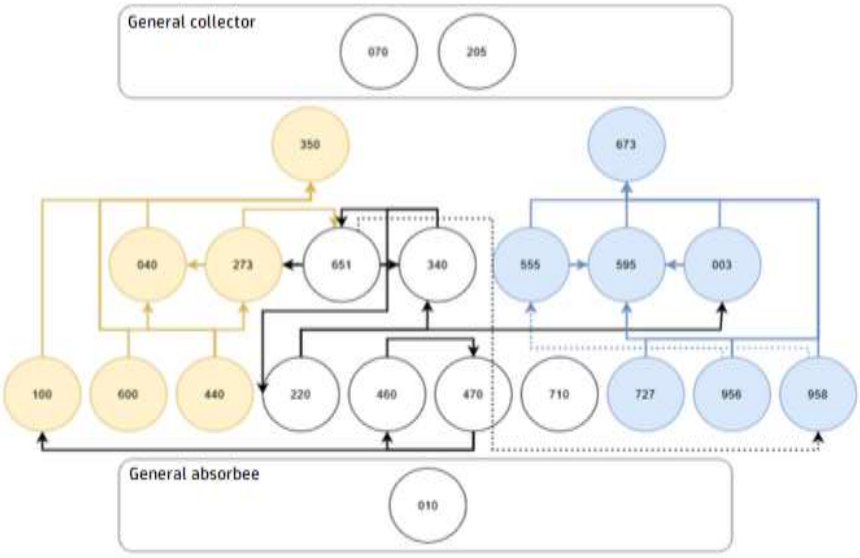


Figure: Prediction relation graph between the function relator codes

We notice the apparition of two specialised ‘collective bins’, the classes *350 – Engraver* and *673 – Director of the research team*. In this case, these classes represent respectively the artistic and academic classes. The existence of these collective bins shows there is not enough features to distinct between similar roles. We have to diagnose the mispredictions and determine if and how we can better our results.

5 – Further work

Firstly we wish to optimise and improve on the results achieved and work accomplished. There are over a hundred function relator codes and we were able to implement a text classification model able to predict only for 33 function relator codes. And we were able to obtain good enough results to justify an eventual implementation of our models on a grand scale for only 13 function relator codes among the 33 function relator codes.

To achieve this, we plan to vastly increase the amount of data we can allocate to our text classification model's training. As we are currently constrained by the issue of keeping our classes balanced, we can both increase the amount of bibliographical records extracted and used to automatically create training entries and/or manually extract records including the lacking function relator codes to create specific training entries.

We also plan to increase the quantity of features that a model may rely upon when making its decision to increase the quality of the prediction and better differentiate between similar function relator code. We are currently working hand in hand with a team of cataloguing experts to analyse the false predictions and explore new potential features.

Another possible solution would be to create different models for different needs. To elaborate, we could create and train multiple models that are more specialised to a certain category of documents rather than using a large all-encompassing model. This would allow us to both reduce class constraints when it comes to creating balanced training sets but also allow our models to be able to better differentiate between similar role codes as it will be able to focus on the finer details and differences.

We can also implement different classification algorithms to find one better suited to our needs and not limit ourselves to the KNN algorithm.

And we may also implement an additional decision making algorithm above the current prediction model to increase the accuracy. The aim would be to analyse the score given by the initial prediction model and only act upon it at a certain threshold. To present this idea in simpler terms, we would only take into consideration the prediction if it is 'sure enough' and ignore it if it is 'unsure'. This would increase precision at the cost of recall, but this is a sacrifice we would be willing to take given the nature of the project. This would also give us the option to accept multiple predictions instead of the initially single one offered.

We would also want to take into consideration the hierarchy between roles. For example, an engraver (350) is a more specific type of author (070).

Finally we have to implement the generation of missing access points. It requires to differentiate between first and last names to 1/ Use the logical rules based system tool Qualinka[10], to link the person names to their idRef identifier. 2/ If it cannot be linked, we will still need to differentiate first and last names when creating the UNIMARC zones and subzones for the access point.

6 – Conclusion

The objective of this project is to teach models to analyse the text of statement of responsibility to automatically generate missing access points for contributors - or adding the function relator code when missing in an existing access point.

It first requires the extraction of two types of entities: persons and roles. The standard Spacy model used for this NER task gives excellent results (precision > 0.99 and recall > 0.92) after being re-trained on 10 000 manually annotated records. This annotation task, critical but often time consuming, was efficiently achieved with the help of a dedicated annotation web tool. Within this comfortable environment, AI assisted librarians can produce more reliable training data which will result with further improvement of the AI.

When the persons and roles are extracted and paired, the next arduous task is to find the right function relator code among UNIMARC relators. Our first results with KNN as classification algorithm will be completed and hopefully enhanced in various ways: more data, more features, more models (specialised), more algorithms, … and more librarians to analyse the predictions and work with the data scientist.

This project is still a work in progress. If the end result is satisfactory, it could be used in production to create new access points or check if the existing ones are correct. But no matter the final conclusion of the project, it will have taught Abes a lot: 1/ As a bibliographic agency and a massive "data steward", Abes has the duty to quickly adopt the machine learning approaches to fulfil its traditional missions ; 2/ with the help of efficient tools to annotate and prepare the data and with librarians in the loop, we can achieve real progress both in terms of data quality and human resource development, two main issues for libraries.

10 Le Provost, Aline and Nicolas, Yann. "IdRef, Paprika and Qualinka**. A toolbox for authority data quality and interoperability**" *ABI Technik*, vol. 40, no. 2, 2020, pp. 158-168. https://doi.org/10.1515/abitech-2020-2006

Acknowledgments

The authors are grateful to Pascal Poncelet (003), Abes colleagues who helped us to extract and analyse data, and Stephen Finnigan.