Capstone proposal by Andreas Jud

Deduplication of Swissbib raw data

Author Andreas Jud</p> **Date** 14-MAR-2020

Task The details are outlined in the **Proposal** unit on the platform - you should address all points from those instructions with as many markdown/code cells as needed. This should include code, observations, discussions and the planned steps.

Introduction

Swissbib is the online catalogue of all Swiss university libraries, the Swiss National Library, numerous cantonal libraries, and further institutions. It provides a fast, simple, and extensive access to scientific publications in Switzerland [1].

1) The problem

Swissbib receives electronic catalogue data of Swiss library network databases with more than 800 institutions behind. The data is processed with dayly delta loads. The Swissbib platform merges the received data records from all sources with the data records received with earlier loads. After that, the platform processes all records into its search engine, where users can search for bibliographic entities. The key step in this merging process is to identify duplicate physical entities out of the original raw data with the goal of combining their records, generating a set of unique records as the basis for indexing in the search engine. This deduplication process has to handle records with minor differences referring to copies of the same physical entities in two different libraries.

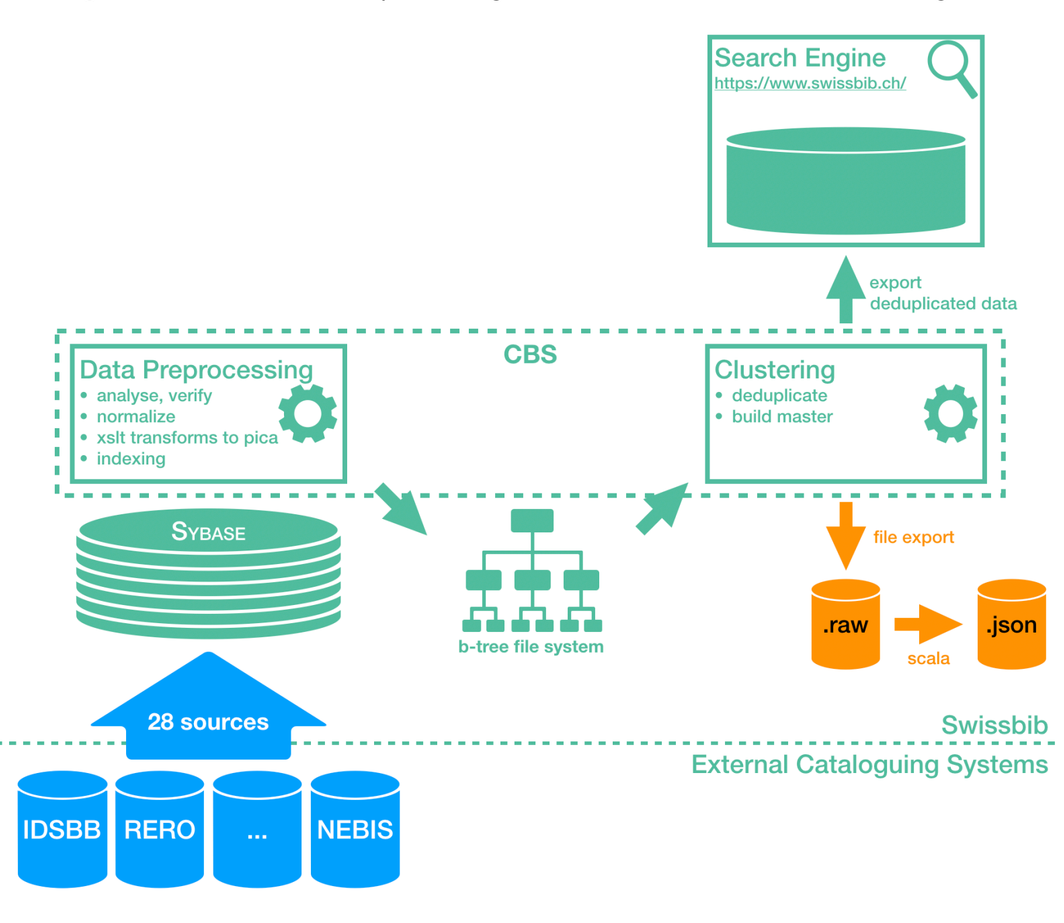
Swissbib's data processing platform CBS transforms data from its sources into a data format called pica, see Data Preprocessing in figure 1, and deduplicates these indexed records with an engine written in programming language C and configured with pica, see Clustering in figure 1. The identification of duplicate records in the data records is done with the help of a set of predefined attributes. The records to be deduplicated are compared in these attributes with a sophisticated logic. Goal of the capstone project is to replace Swissbib's established deduplication logic with the help of Machine Learning.

2) The data

(a) Clear overview of your data

The Swissbib platform loads raw data out of its sources in dayly delta loads and stores all records in a Sybase database, see figure 1. After having preprocessed all data records from its sources, a module of the CBS platform indexes the raw data into a b-tree file system. This indexed data is then forwarded by a clustering process to an index of the search engine. This data is in a deduplicated state and is available online [1] for end users to search for catalogue records and its physical entities behind.

Figure 1 Basic data flow for processing Swissbib's raw data into its search engine.



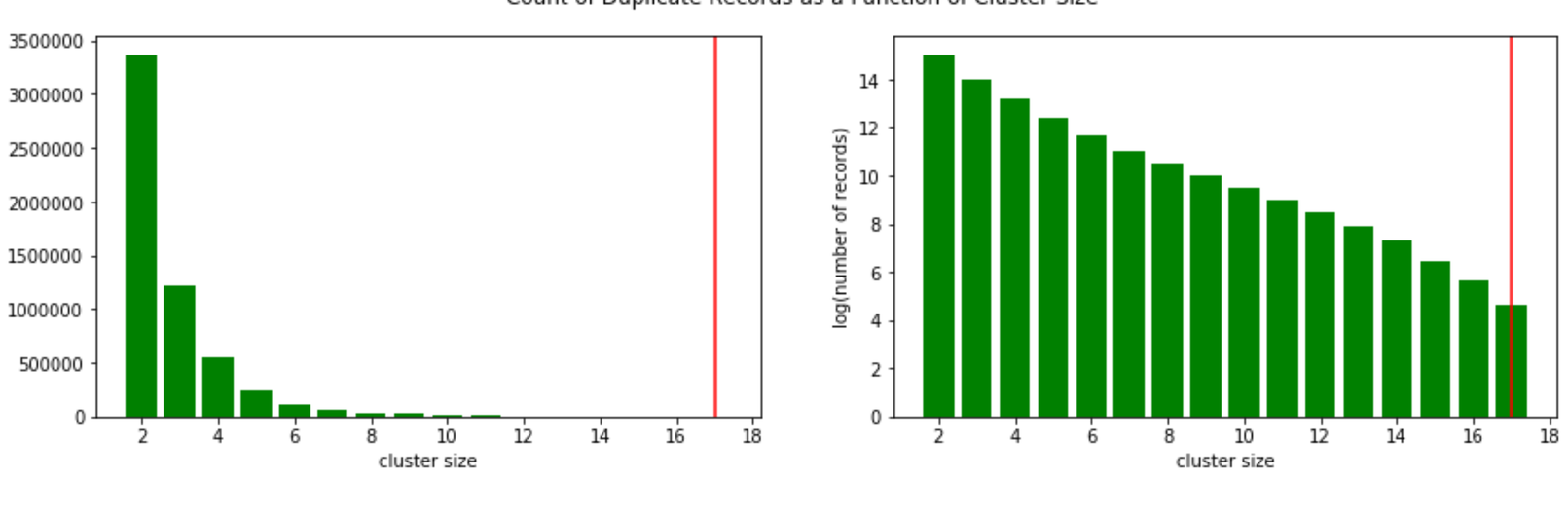
The Swissbib platform holds records of a total number of more than 31 million distinct entities out of 28 source databases [2]. The raw data of the sources is deduplicated in a clustering module of the CBS platform, see figure 1, in order to identify the set of records describing distinct physical entities. The data extract below [2] illustrates the amount of data being held and processed by the platform. In the data extract shown, a maximum threshold of 17 records per cluster is considered to be a reliable amount [2].

In [1]:
```python
import pandas as pd

df = pd.read_csv('./data/cluster_sizes.csv', sep='\s+',
                 header=None, names=['number_of_clusters', 'records_in_cluster'])

cluster_size_threshold = 17
df.where(df.records_in_cluster <= cluster_size_threshold).dropna().astype(int)
```

Out[1]:

	number_of_clusters	records_in_cluster
0	3369004	2
1	1212276	3
2	545500	4
3	241145	5
4	112948	6
5	60816	7
6	35637	8
7	21578	9
8	13336	10
9	7932	11
10	4662	12
11	2728	13
12	1453	14
13	641	15
14	281	16
15	105	17

In [2]:
```python
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

fig, axs = plt.subplots(1, 2, figsize=(15, 4))
axs[0].bar(df['records_in_cluster'].where(df.records_in_cluster <= cluster_size_threshold).dropna().astype(int),
           df['number_of_clusters'].where(df.records_in_cluster <= cluster_size_threshold).dropna().astype(int)
        ,
        color='green')
axs[0].axvline(x=cluster_size_threshold, color='red') # Threshold
axs[0].set_xlabel('cluster size')
axs[0].set_ylabel('number of records')
axs[1].bar(df['records_in_cluster'].where(df.records_in_cluster <= cluster_size_threshold).dropna().astype(int),
        np.log(
           df['number_of_clusters'].where(df.records_in_cluster <= cluster_size_threshold).dropna().astype(int)
        )
        ,
        color='green')
axs[1].axvline(x=cluster_size_threshold, color='red') # Threshold
axs[1].set_xlabel('cluster size')
axs[1].set_ylabel('log(number of records)')
fig.suptitle('Count of Duplicate Records as a Function of Cluster Size')
plt.show()
```



The amount of processed records of the Swissbib platform as of date of [2] sums up to the structure below.

In [3]:
```python
number_of_uniques = 25479651
number_of_masters = df.number_of_clusters.where(df.records_in_cluster <= cluster_size_threshold).sum().astype(int)
number_of_slaves = (df.number_of_clusters
#                    .where(df.records_in_cluster <= cluster_size_threshold)
                   *(df.records_in_cluster
#                      .where(df.records_in_cluster <= cluster_size_threshold)-1
                    )).sum().astype(int)

print('Count of records ...')
print('- original\t{:,} (100%)'.format(number_of_uniques+number_of_masters+number_of_slaves))
print('... and thereof ...')
print('- duplicates\t{:,} ({:.1f}%)'.format(number_of_slaves,
                                        100*number_of_slaves/(number_of_uniques+number_of_masters+number_of_slaves)))
print('- uniques\t{:,} ({:.1f}%)'.format(number_of_uniques,
                                    100*number_of_uniques/(number_of_uniques+number_of_masters+number_of_slaves)))
print('- clusters\t{:,} ({:.1f}%)'.format(number_of_masters,
                                    100*number_of_masters/(number_of_uniques+number_of_masters+number_of_slaves)))
print('\n- fully deduplicated\t{:,} ({:.1f}%)'.format(number_of_uniques+number_of_masters,
                                                100*(number_of_uniques+number_of_masters)/(number_of_uniques+number_of_masters+nu
mber_of_slaves)))
print('\t\t\t==========')
```

```
Count of records ...
- original      46,816,243 (100%)
... and thereof ...
- duplicates    15,706,550 (33.5%)
- uniques       25,479,651 (54.4%)
- clusters      5,630,042 (12.0%)

- fully deduplicated    31,109,693 (66.5%)
                        ==========
```

With the sru interface [3], the Swissbib project offers a protocol to acces its processed and indexed data via URL. The search engine can be queried via this interface to deliver masses of records in an xml or a json format responding to the sent request. The data via [3] partially underlies open acces according to the Creative Commons license of CC0 [4].

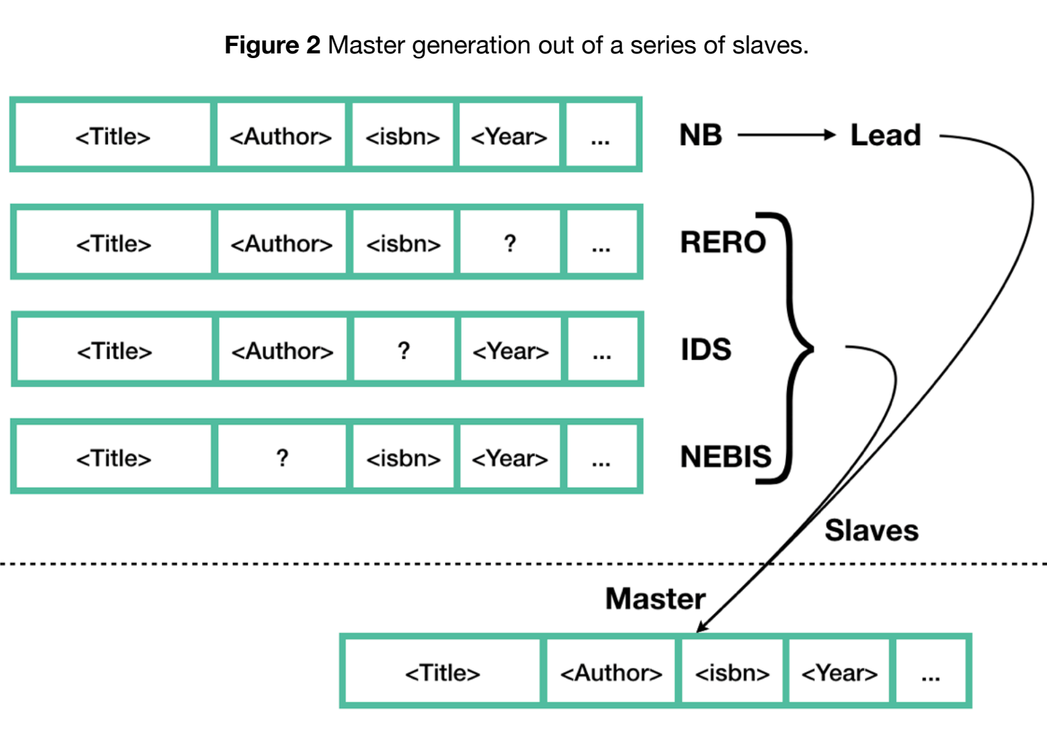
Some parts of the data of [3] is under protection, though, e.g. records enriched with data from worldcat. These parts of the data will be filtered out of the data to be processed during the capstone project. This will be guaranteed by a specific preprocessing of the data for the capstone project, compare orange coloured data path of figure 1 and description below. The search engine [5] offers access to the fully open data, exclusively.

A data record provided by Swissbib consists of a set of data elements filled with their values. The structure and contents of the data records are based on MARC 21 Format for Bibliographic Data, [6]. Within the scope of the capstone project, the data records will be narrowed down to a reduced set of the available attributes of the MARC 21 standard. The reduction focusses on the attributes that are used for deduplication with the already implemented logic. This reduction is done in a separate data export with a scala implementation, developped by a Swissbib project member specifically for the capstone project [7], see orange coloured part of figure 1. The resulting output of the data preprocessing by Swissbib will be one (or more) file(s) in a json format. This file will be read into Python for exploratory data analysis (EDA) and manipulation in the scope of the capstone project.

The list of attributes that will be used as a starting point for the capstone project is documented in [8]. Some additional attributes out of the MARC 21 attribute definition may be identified and added optionally during the project. This option can be taken if additional attributes help to enhance the deduplication process.

The existing conventional deduplication implementation searches for duplicate records in the raw data. This is done with a sophisticated comparison logic of the above mentioned attributes. If two or more records are found to describe the same bibliographical entity, one of the records is identified as the lead record. Afterwards, all of the identified duplicate records are united in a single record. This record is called *master*. The original records become *slaves* of the master record. Eventually missing or differing data fields are consolidated into the master record. This process is sketched in figure 2.

Figure 2 Master generation out of a series of slaves.



The Swissbib project team holds a set of pre-defined records they use for testing purposes of their deduplication implementation. This set of records is called the *goldstandard* and will be used as the data set for training and performance testing during the machine learning process.

Swissbib's goldstandard data comes along in three json files. Files `slave.json` and `unique.json` hold records in their original form. While each record in file `unique.json` represents a unique bibliographical entity, the records in file `slave.json` represent a varying number of duplicate bibliographical entities. One of the three generated files is called `master.json`. It holds all master records of the records of file `slave.json` and each record is the unique representation of duplicate bibliographical units. The counts of the records in each file are calculated below.

In [4]:
```python
import json as js

records_masters, records_slaves, records_uniques = [], [], []

for line in open('./daten_goldstandard/master.json', 'r'):
    records_masters.append(js.loads(line))
for line in open('./daten_goldstandard/slave.json', 'r'):
    records_slaves.append(js.loads(line))
for line in open('./daten_goldstandard/unique.json', 'r'):
    records_uniques.append(js.loads(line))

print('Count of records in file ...')
print('- slave\t', len(records_slaves))
print('- master', len(records_masters))
print('- unique', len(records_uniques))
```

```
Count of records in file ...
- slave  435
- master 159
- unique 596
```

For the machine training and testing process, pairs of records will be generated, see below. Each pair of the goldstandard will be marked whether it is a pair of duplicates or a pair of uniques.

The amount of available records for training and performance testing out of the goldstandard depends on the explicit number of duplicate records in file `slave.json` and cannot be calculated in advance. A rough estimate can be calculated with the formula $\frac{1}{2}N\,(N-1)$ where N is the total number of masters and uniques of the goldstandard data set.

In [5]:
```python
uniques_masters_sum = len(records_masters)+len(records_uniques)

print('{:,}'.format(int(uniques_masters_sum*(uniques_masters_sum-1)/2)))
```

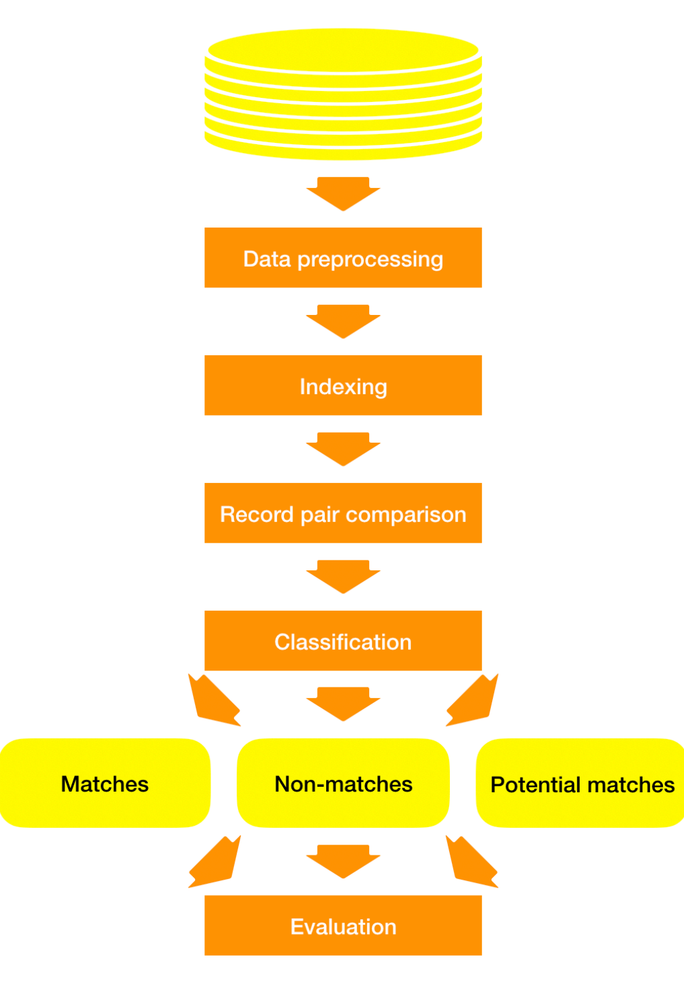
```
284,635
```

(b) Plan to manage and process the data

The capstone project will start with the analysis of exported data by the orange data extract of figure 1. Goal will be to understand the data and the relationships in its attributes. The data comes along with substantial information in form of strings, e.g. in title and person, see below. Besides string manipulations, the exploratory data analysis will address handling of missing values and decide on actions to be taken for data cleaning.

A general data matching process is described in [9], see figure 3. Swissbib's CBS platform has implemented this process.

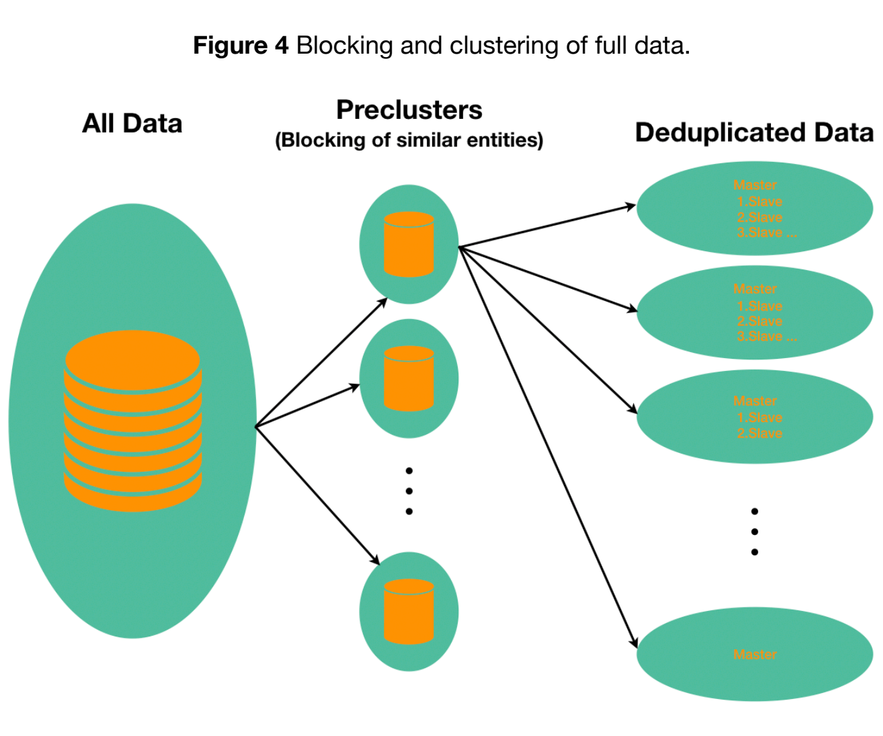
Figure 3 General data matching process [9].



A data preprocessing step will be implemented in the beginning of the capstone project. The data will be manipulated in this step with the goal of generating a set of feature attributes for the machine learning process of deduplication. The features will drive the machine learning and will enable the final deduplication process.

After the data preprocessing step in the general data matching process shown in figure 3, the indexing of the data is a central step in preparing the data for record pair comparison [9]. During indexing, the full data set is split into blocks that hold records of similar entities. This splitting is called blocking [9] and the resulting blocks are called preclusters. Preclusters are the starting point for explicit record comparison in the deduplication process, see figure 4. The indexing step is crucial in that a minimal size of the preclusters reduces the processing time of the record pair comparison.

Figure 4 Blocking and clustering of full data.



The indexing and preclustering is done today by the CBS platform, see figure 1. The Swissbib project team plans to replace this indexing step with a new platform, see section 4c) below. The preclustering is outside of the scope of the capstone project.

After indexing, the record comparison is done pairwise for all records in each precluster, see figure 3. In the record pair comparison step, the predefined attributes of two records are compared and a distance between them is derived. This step is the starting point of the central part of machine learning of the capstone project as the set of features to be used for model training will be specifically defined, see below. The data processing of the capstone project ends with the identification of the specific features for the machine lerning process and the definition of the distance metrics for each attribute, see below.

3) Exploratory data analysis (EDA)

Below, a sample data set of Swissbib's data extract is shown.

In [6]:
```python
df_s = pd.DataFrame(records_slaves)

# Extend display to number of columns of DataFrame
pd.options.display.max_columns = len(df_s.columns)

df_s.head(10)
```

Out[6]:

	docid	035liste	isbn	ttlfull	ttlpart	person	corporate	pubyear	decade	century	exactDat
0	000311049	[(OCoLC)731635279, (ABN)000539983]	[978-3-15-020008-7]	{'245': ['Emma', 'Roman']}	{'245': ['Emma', 'Roman']}	{'100': ['AustenJane'], '700': ['GraweChristia...	{'110': [], '710': [], '810': []}	2009	2009	2009	200
1	00130724X	[(OCoLC)808324878, (ABN)000155059]	[]	{'245': ['Die Zauberflöte', 'Oper in zwei Aufz...	{'245': ['Die Zauberflöte', 'Oper in zwei Aufz...	{'100': ['LevineJamesDir.'], '700': ['MozartWo...	{'110': [], '710': ['Metropolitan Opera Orches...	2000	2000	2000	200
2	001817272	[(OCoLC)231772550, (ABN)000096920]	[3-495-47879-5]	{'245': ['Der moralische Status der Tiere', 'H...	{'245': ['Der moralische Status der Tiere', 'H...	{'100': ['FluryAndreas'], '700': [], '800': []...	{'110': [], '710': [], '810': []}	1999	1999	1999	199
3	00236865X	[(OCoLC)887157168, (ABN)000223912]	[]	{'245': ['Die Zauberflöte']}	{'245': ['Die Zauberflöte']}	{'100': ['MozartWolfgang Amadeus'], '700': [],...	{'110': [], '710': [], '810': []}	uuuuuuuu	uuuu	uuuu	uuuuuuu
4	00351031X	[(OCoLC)887324690, (ABN)000548154]	[978-1-4058-8214-9]	{'245': ['Emma']}	{'245': ['Emma']}	{'100': ['AustenJane'], '700': ['StrangeJoanna...	{'110': [], '710': [], '810': []}	2008	2008	2008	200
5	003517837	[(OCoLC)180154028, (ABN)000308006]	[978-3-7815-1531-4, 3-7815-1531-1]	{'245': ['Lernprozesse dokumentieren, reflekti...	{'245': ['Lernprozesse dokumentieren, reflekti...	{'100': [], '700': ['Gläser-ZikudaMichaela'], ...	{'110': [], '710': [], '810': []}	2007	2007	2007	200
6	003555208	[(OCoLC)605622457, (ABN)000234417]	[]	{'245': ['Die Zauberflöte']}	{'245': ['Die Zauberflöte']}	{'100': ['MozartWolfgang Amadeus'], '700': ['H...	{'110': [], '710': [], '810': []}	1988	1988	1988	198
7	003959953	[(OCoLC)887393628, (ABN)000218626]	[]	{'245': ['Die Zauberflöte']}	{'245': ['Die Zauberflöte']}	{'100': ['MozartWolfgang Amadeus'], '700': [],...	{'110': [], '710': ['Opernhaus Orchester Züric...	2002	2002	2002	200
8	005047420	[(OCoLC)778329562, (ABN)000048149]	[0252-9955]	{'245': ['Bildungsforschung und Bildungspraxis']}	{'245': ['Bildungsforschung und Bildungspraxis']}	{'100': [], '700': [], '800': [], '245c': ['hr...	{'110': [], '710': ['Schweizerische Gesellscha...	19791999	1979	1979	1979199
9	005396816	[(OCoLC)780137741, (ABN)000669941]	[978-3-7255-6535-1]	{'245': ['Sozialleistungsbetrug - Sozialversic...	{'245': ['Sozialleistungsbetrug - Sozialversic...	{'100': ['KäserBeatrice'], '700': [], '800': [...	{'110': [], '710': [], '810': []}	2012	2012	2012	201

(a) Preliminary EDA

A first sample data delivery by Swissbib's project team in the form of a json file with a number of 183'407 records dating from 10.11.2019 has revealed information of the attributes that is summarized in a table of Swissbib's Wiki pages [8]. These attributes cover a wide range of data types like strings of text, codes, and numbers in the form of lists. On the other hand, the data shows missing values in most of the attributes.

In a first step, the quality of the data extract for the capstone project, see orange coloured data path of figure 1, has to be verified and improved to a useful level [7]. This will be done by the following activities.

- Verifying random sample data to the results of a data search on Swissbib's web site [1].
- Investigating missing values and the degree of filling of the attributes.
- Refining the structure of the data, see below.
- Checking for plausibilities with the help of relationships between features.
- Checking for outliers.

In a first basic analysis, done with a very early data extract, the attributes `ttlfull` and `person` were found as a list of comma-separated strings originating in combinations of several parts of the MARC 21 data strucutre, see [8]. This superstructuring has been questioned and reengineered afterwards in a detailed form, shown in the data extract of the goldstandard above. This kind of data analysis and structure refinement will be elaborated in the beginning of the capstone project.

(b) How does the EDA inform your project plan?

The important step of the deduplication process is record pair comparison in figure 3. The challenge of this step is to find the attribute set with the best effect on duplicate prediction together with some distance metrics for a clear identification of pairs of duplicate records. This challenge will be taken by testing several similarity metrics, like e.g.

- Levenshtein edit similarity
- some q-gram based similarity
- Jaccard similarity
- etc.

[9], and assessing their effect on the prediction quality of the model. The best distance metric(s) found will be applied on the defined set of attributes of each pair of records to be compared, resulting in a set of new, derived attributes for comparison. These new attributes of a record pair will be the features of the machine learning model.

(c) What further EDA do you plan for project?

Swissbib data contains more attributes than the list for deduplication [8]. The degree of filling and the quality of these potential attributes is unknown, at the moment. One task might be to identify additional promising attributes to enhance the performance of the trained model. This will be done with the help of the MARC 21 documentation [6]. If such promising additional attributes can be identified, the Swissbib project team will be asked to add them to their data extract.

4) Machine learning

The last step of a general data matching process is classification, see figure 3. In this step of the capstone project, machine learning comes into play. It will be used for calculating the thresholds between matching, potentially matching, and non-matching record pairs. The records will be classified with the help of different methods, see below. Swissbib's test data of the goldstandard will allow training of the model during this step.

The performance of the model will be measured either with a fraction of the training data or with additional deduplicated data of the Swissbib platform. If a fraction of the training data will be used, this fraction will be excluded from the training of the model. The key indicators out of the confusion matrix will be used for assessing the performance of each model analysed.

(a) Phrase your project goal as a clear machine learning question

The goal of the capstone project is to implement and train an artificial neural network for deduplication [10] of library catalogue data in MARC 21 format. The neural network will be trained with Swissbib's goldstandard data for deduplication. The performance of the resulting model will be analysed with some key indicators derived from the confusion matrix.

(b) What models are you planning to use and why?

The following machine learning models will be used in the scope of the capstone project.

- [9] suggests deduplication with the help of decision trees. The capstone project will analyse the outcomes of a decision tree model.
- [9] suggests deduplication with a support vector machine (svm). The results of an svm model with the model above will be compared.
- [10] describes a deduplication method based on an artificial neural network. The approach of this reference will be tried to reproduce for the given data.

(c) Please tell us your detailed machine learning strategy

If the accuracy of the prediction of duplicates is comparable with the implemented conventional deduplication logic then the implemented deduplication will be replaced with the new model in the context of an Apache Flink or an Apache Spark implementation.

5) Additional information

References

[1] - https://www.swissbib.ch/, Access web site of Swissbib's online catalogue.
[2] - Communication by Silvia Witzig, Universität Basel, Universitätsbibliothek, Projekt swissbib, 15.10.2019.
[3] - http://sru.swissbib.ch/, Interface to Swissbib's online catalogue as Search/Retrieve via URL.
[4] - https://creativecommons.org/share-your-work/public-domain/cc0/, Creative Commons CC0 license documentation.
[5] - https://data.swissbib.ch/, Documentation of Swissbib's CC0 open data.
[6] - https://www.loc.gov/marc/bibliographic/, MARC 21 format description for bibliographic data, Library of Congress
[7] - https://github.com/guenterh/andreas_cluster_features, Repository to scala code for data extraction.
[8] - http://www.swissbib.org/wiki/index.php?title=Features_Deduplication, Wikipedia site created for documentation of features for deduplication.
[9] - *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Peter Christen, Springer-Verlag 2012.
[10] - *An approach based on artificial neural network for data deduplication*, M. Padmanaban, T. Bhuvaneswari, International Journal of Computer Science and Information Technologies, Vol. 3(4), 2012, 4637-4644.

In [ ]: