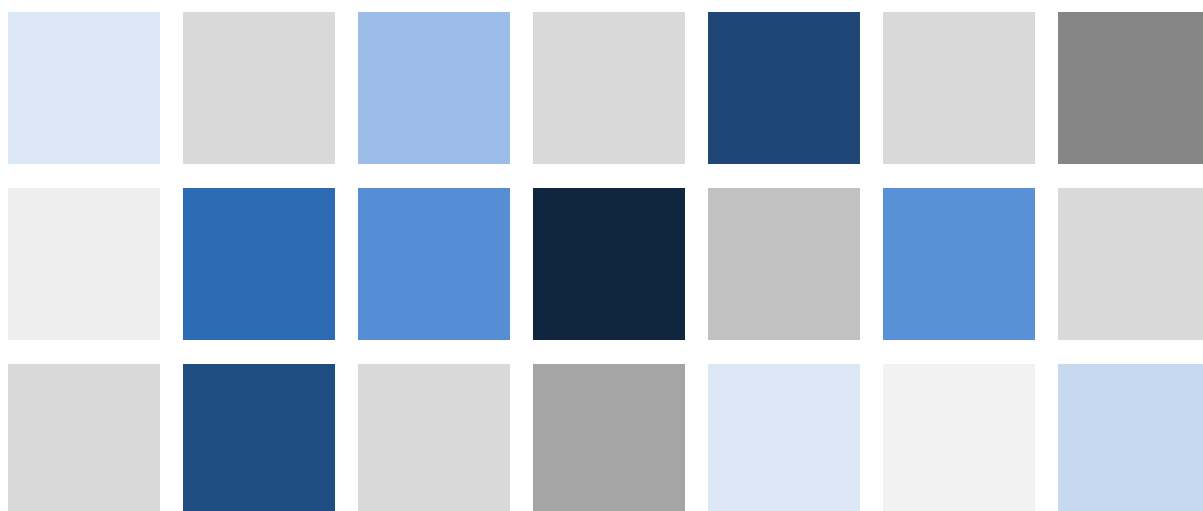


Long-term data for Europe

# EURHISFIRM

## M6.1: Data Matching Case Study



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 777489

<https://eurhisfirm.eu>

**AUTHOR(S):**

Boris CULE (University of Antwerp)

Frans BUELENS (University of Antwerp)

Johan POUKENS (University of Antwerp)

Jan ANNAERT (University of Antwerp)

Johan RICHER (with the Paris School of Economics)



## Table of Contents

Introduction.....	4
Case 1: Data Matching within the National Databases.....	4
Experimental Design.....	4
Matching Corporations.....	5
Data Pre-processing.....	5
Exact Matches on Corporation Name .....	5
Distance Measures .....	6
Non-exact Matches .....	7
Matching Securities .....	11
Security Name Matching .....	11
Stock Price Matching .....	12
Stock Dividend Matching.....	13
Bond Price Matching .....	13
Case 2: Developing a Collaborative Environment .....	14
Context .....	14
Wikibase .....	15
Experiment with DFIH and SCOB data and first results.....	17
Data .....	17
Steps .....	18
Import entities.....	18
Reading and editing an item.....	19
Matching.....	21
Merging .....	21
Exploitation.....	25
Conclusion .....	29
References .....	29



## Introduction

In this report, we present the results of two data matching case studies. The first study, performed by the Antwerp team, focused on directly matching the data currently present in two of national databases available to the project. The second study, performed by the Paris team, focused on developing a collaborative environment to facilitate registering data matching results within the Wikibase format.

## Case 1: Data Matching within the National Databases

The goal of the study was to compare and evaluate various data matching techniques, in order to see which ones are feasible within the EURHISFIRM project, and, of those, which ones perform best.

We begin the report by describing the experimental design, before moving on to discussing the necessary pre-processing steps, and, finally, reporting the results of each individual technique and comparing them to each other. We conclude with the most important lessons learned from the case study.

### Experimental Design

In order to keep the size of the experiment manageable for human supervision, we limit the case study to matching the data originating in the SCOB and DFIH databases. Furthermore, we focus on the time period between 1 January 1890 and 31 December 1906. Not only is this a period that both the Paris and Brussels stock market were well populated, but it also straddles new French regulation introduced in 1898. Indeed, in this year the French government approved a regulation improving the transparency of both the Paris Bourse and the Paris OTC market (Hautcoeur & Riva, 2012). By selecting this period, it thus becomes possible to study the impact this regulation had on market quality and the interconnections between the Paris Bourse, the Paris OTC market and the Brussels Stock Exchange.

The case study consists of two main phases. In the first phase, we attempt to match corporations to each other, and in the second phase, we do the same for individual securities (stocks or bonds).

To this end, we evaluate a number of different data matching techniques, while at the same time examining the quality of the available data and the usefulness of various data items for this task. The performance of the techniques is mainly evaluated using the true positive rate, or the ratio between correctly identified matches and the total number of identified possible matches. The true positive rate is computed using human verification of the possible matches. Note that we have no way of computing the false negative rate, since we have no prior knowledge of all the matches that exist in the two databases.



## Matching Corporations

In this section, we describe the first phase of the case study, in which we attempt to identify which corporations can be found in both SCOB and DFIH databases.

### Data Pre-processing

Before we could begin with the data matching task, some data pre-processing was necessary.

First of all, we identified which data could be of use for the matching task. To begin with, we focus on corporation names, as well as the start and end date of particular names. We run an SQL query in each database that produces all corporation IDs, as well as their names, start and end dates, that had a security listed at the respective stock exchange at some point during the period of interest (between 1 January 1890 and 31 December 1906). This query retrieved 2211 entries from the SCOB database, and 3565 entries from the DFIH database. This output was then exported into csv files for further use by the data matching algorithms.

After a manual inspection, we noticed some inconsistencies in the data, and, to obtain better results, we replaced all instances of ' with '. Note that, during this case study, no changes have been made to the underlying databases themselves, but we nevertheless report all lessons learned that could lead to improvements in data quality in the original databases. Naturally, the performance of any data matching techniques relies heavily on the quality of the available data.

Once we exported the relevant data into csv files, all further experiments, reported below, were implemented and performed in Python.

### Exact Matches on Corporation Name

As a first experiment, we identified corporations that have exactly the same name in the two databases. This produced 30 corporation names. For illustration, we provide a subset of the output in the table below:

EXAMPLE	NAME	SCOB.ID	DFIH.ID	SCOB.START	SCOB.END	DFIH.START	DFIH.END
1	Banque de Flandre	285	1068	13 Aug 1841	03 Nov 1928	21 Aug 1841	31 Dec 3999
2	Société Métallurgique de Couillet	1353	3255	22 Mar 1906	31 Dec 3999	01 Jul 1835	31 Dec 3999
3	Société des Sels gemmes et Houilles de la Russie méridionale	9485	2427	15 Jun 1883	31 Dec 3999	31 Dec 1896	31 Dec 3999



The full output (30 corporation pairs) has then been provided to a domain expert (Frans Buelens of the University of Antwerp) for verification. Surprisingly, even the simplest technique imaginable (exact match on corporation name) produced one false positive. The second example given in the table above was not a true match. One of the two corporations was in fact liquidated in 1881, while the other was renamed in 1906. While there exist a link between the two, they are clearly not the same and should not be matched. Based on this, we conclude that, regardless of the method used, all potential matches must be inspected and verified by a human expert. Nevertheless, this first experiment produced a true positive rate of  $29/30 = 96.67\%$ .

An important conclusion from this experiment was the fact that the start and end dates were too unreliable to be used by any automated technique. In very few cases, both the start and the end date actually matched. In other cases, the dates were similar (e.g., start date in the first example above), but in others they were considerably different (e.g. start date in the third example above). Furthermore, many of the dates are not even there, and dummy dates are used instead (e.g., 31 December 3999, which is used as end date when the date is not known, but also in many cases where the corporation is still active). We conclude that start and end dates should not be used for automatic record matching, but we provide them to human experts to potentially facilitate verification (these could also be used for correcting or enriching the data in the underlying databases, but this is beyond the scope of this case study). Finally, for every identified match, we need to note the period in which the two corporation IDs match. As a safe choice, as the start date of the match we take the latest of the two start dates in the original databases, and, as the end date of the match we take the earliest of the two end dates.

### Distance Measures

Naturally, most matches in the data cannot be identified using exact matching. This can be due to variations in names, languages, spelling, or even simple typos. Therefore, in our further experiments, we attempt to measure the distance (or, conversely, similarity) between two corporation names in order to identify further matches. In short, the more similar two names are, the more likely it is that they form a match.

In text analysis, the two most commonly used techniques are the Jaro-Winkler similarity (Winkler, 1990) and the Levenshtein distance (Levenshtein, 1966). The main difference between the two techniques is that Jaro-Winkler similarity gives more weight to differences at the start of the strings (in our case corporation names) than to those near the end, while the Levenshtein distance gives equal importance to differences anywhere within the strings.

We first performed some tests using the Jaro-Winkler similarity (imported from the `pyjarowinkler` package in Python). Surprisingly, when looking for matches with similarity equal to 1, the output contained 56 pairs of corporation names. However, similarity can only be equal to 1 if the match is exact, and our previous experiment discovered just 30 exact matches. Upon inspection, it turned out that this particular implementation of the Jaro-Winkler similarity is case-insensitive, meaning that lowercase and uppercase versions of the same letter were considered to be equal. A few examples of the newly-discovered matches are provided below (we omit the start and end dates):

SCOB.ID	SCOB.NAME	DFIH.ID	DFIH.NAME
187	Fabrique de Fer d'Ougrée	1341	Fabrique de Fer d'ougrée
3414	Société Minière et Industrielle de Routchenko	3238	Société minière et industrielle de Routchenko
11514	Banque hypothécaire franco-argentine	3047	Banque Hypothécaire Franco-Argentine

Given this insight, we repeated our first experiment for exact matches, but having first converted all corporation names into lowercase. By doing this, we obtained the same list of 56 matches as above. Finally, we searched for all pairs of corporation names with a Levenshtein distance of 0, again expecting to find all exact matches, and discovered that the Levenshtein distance (imported from the `editdistance` package in Python) was case sensitive, producing only the 30 matches found in our first experiment. Recomputing the Levenshtein distance on lowercase names produced the same 56 matches as above.

Upon inspection by a human expert, it turned out that all new cases were in fact true positives, giving us a true positive rate of  $55/56 = 98.21\%$ . Additionally, the matches obtained in this way could also prove to be a valuable tool for data cleaning, and could lead to more uniformity in corporation names across the two databases.

### Non-exact Matches

Having examined the exact matches found in the data and concluded that the true positive rate was satisfactory, but not perfect, we moved on to trying to identify non-exact matches. First, we used the Jaro-Winkler similarity, with the filtering threshold set to 0.95. Naturally, the lower the similarity, the more uncertain the match. Our search yielded 108 possible matches (including the 56 exact matches). We list a few new examples in the table below:

EXAMPLE	SCOB.ID	SCOB.NAME	DFIH.ID	DFIH.NAME	SCORE
1	615	Caisse d'annuités dues par l'Etat (BELGE)	1808	Caisse d'Annuités dues par l'Etat	0.96
2	933	Compagnie des Chemins de fer de l'Est	1606	Compagnie des Chemins de Fer de l'Est Algérien	0.96
3	1221	Compagnie des Tramways de Reims	1623	Compagnie des Tramways de Nantes	0.96
4	1221	Compagnie des Tramways de Reims	2644	Compagnie des Tramways de Rouen	0.96
5	3447	Providence Russe (à Marioupol)	3277	Providence Russe a Marioupol	0.97

Even at first glance, it was obvious that the new list contained a lot more false positives than exact matches only. Furthermore, for some corporation names the method identified multiple possible matches (e.g., examples 3 and 4 above). The fact that the Jaro-Winkler similarity gave more weight to the beginning of the string proved useful in some cases (e.g., examples 1 and 5 above), but it clearly led to a rise in false positives (e.g., example 2 above). After human inspection, we ascertained that the Jaro-

Winkler similarity above 0.95 yielded a true positive rate of  $90/108 = 83.33\%$ . However, not including the exact matches, the true positive rate among the new examples (i.e., similarity higher than 0.95 and lower than 1) was only  $35/52 = 67.31\%$ , which was a considerable drop.

We ran one further experiment with the Jaro-Winkler similarity. We set the threshold at 0.9, which produced 762 new potential matches (on top of the 108 examined above). However, even at a quick glance, it was obvious that most of these were false positives. Providing such a list to a human expert for manual inspection would have been a waste of resources.

As a result, we turned our attention to the Levenshtein distance. First of all, we considered two variants of this measure. The Levenshtein distance measures how many steps (insert, delete or swap) are required to convert one string into another. The normalised Levenshtein distance computes the same, but relative to the length of the string. Clearly, a distance of 1 in a string of length 4 is more significant than a distance of 2 in a string of length 40. The normalised Levenshtein distance would be 0.25 and 0.05 for the two cases, respectively. Therefore, in all our experiments, we used the normalised Levenshtein distance. Furthermore, we used lowercase versions of the corporation names, which already proved a valuable technique in the exact match search earlier.

To start with, we set the threshold at 0.05, reporting all corporation name pairs with a distance smaller than 0.05. This produced 84 potential matches, including the 56 exact matches. Some of the new examples are listed below:

EXAMPLE	SCOB.ID	SCOB.NAME	DFIH.ID	DFIH.NAME	DISTANCE
1	732	Caisse Générale de Reports et de Dépôts	1480	Caisse Générale de Reports et de Dépôts	0.026
2	10888	Compagnie des chemins de fer de l'Ouest de l'Espagne	2572	Compagnie des Chemins de fer de l'Est de l'Espagne	0.038
3	12848	PROVINCIES : ENTRERIOS (ARGENTINE)	10252	PROVINCES : Entrerios (Argentina)	0.029

First of all, we note that moving the threshold by 0.05 produced fewer new results than doing the same with the Jaro-Winkler similarity. Second, upon human verification, it turned out that the new possible matches contained just one false positive, namely example 2 listed above. However, unlike the false positives produced by the Jaro-Winkler similarity, this example would be tough to avoid using any automated method. Example 3 above once again illustrates the importance of using non-case sensitive methods, while example 1 shows the importance of using diacritics consistently. Furthermore, both these examples show how data matching results can also help improve the correctness, cleanness and uniformity of the original databases. Finally, we conclude this experiment by computing the true positive rate using the normalised Levenshtein distance with the threshold set at 0.05:  $82/84 = 97.62\%$ . Among the new cases (i.e., distance smaller than 0.05 and larger than 0), the true positive rate was  $27/28 = 96.43\%$ .

In our further experiments, we further increased the normalised Levenshtein distance threshold, to see how high we could go, while retaining the high quality of the output. First, we ran the algorithm with a

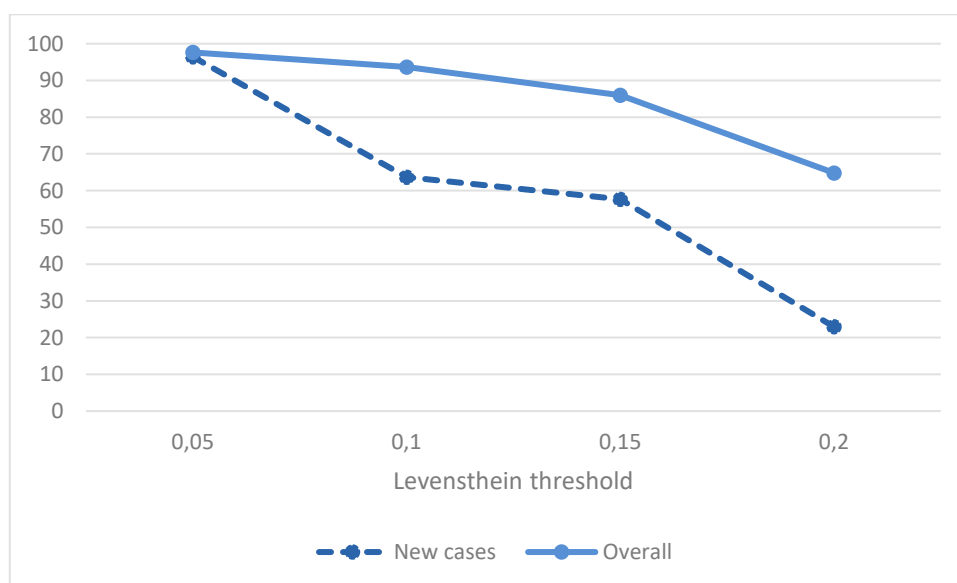


threshold of 0.1, and obtained just 11 new possible matches. However, of those 11, 4 turned out to be false positives. This was, naturally, to be expected – the higher the distance, the more uncertain the possible match. Nevertheless, the overall true positive rate with the threshold set to 0.1 was  $89/95 = 93.68\%$ . Note that this method produced almost the same number of correct matches as Jaro-Winkler similarity with the threshold set to 0.95, but with a considerably smaller number of false positives (and, therefore, with much less human effort). Among the new cases (i.e., those with a distance smaller than 0.1, but larger or equal to 0.05), the true positive rate was  $7/11 = 63.64\%$ .

Next, we increased the distance threshold to 0.15, and discovered another 26 new possible matches. Remarkably, the output still proved valuable, with the overall true positive rate of  $104/121 = 85.95\%$ , and the true positive rate among the new cases (i.e. those with a distance smaller than 0.15, but larger or equal to 0.1) of  $15/26 = 57.69\%$ . We conclude that the true positive rate using a normalised Levenshtein distance with a threshold of 0.15 is still higher than that of Jaro-Winkler similarity with a threshold of 0.95. This is particularly striking since we have now identified 14 more matches than with the Jaro-Winkler similarity, while retaining a relatively high true positive rate.

In a final experiment, we increased the distance threshold to 0.2. This produced 61 new possible matches. However, even a quick glance at the output was enough to see that it included many false positives. This was to be expected, as a distance of 0.2 between two strings is too large to be attributed to simple typos or data input inconsistencies. Nevertheless, of these 61 new cases, 14 did turn out to be true positives (a rate of  $14/61 = 22.95\%$ ). The overall true positive rate at the threshold of 0.2 was  $118/182 = 64.84\%$ , a considerable drop from the earlier experiments.

A summary of the true positive rates at various thresholds is shown in the figure below. Here, “New cases” refers to the potential matches found between the two thresholds (e.g., the point bottom right evaluates only potential matches with the normalised Levenshtein distance smaller than 0.2, but greater than or equal to 0.15), while “Overall” reports the true positive rate for all cases with the distance smaller than the threshold.



At this point, having identified 118 correct corporation matches in the data, we decided to move on to the second phase of the case study – namely, security matching, described in the following section. Clearly, it is quite likely that we haven't identified all the corporation matches in the data, as some could have a distance larger than 0.2. In the long run, it would be ideal to find all correct matches, either through improving the quality of the original data, or by using more specialised algorithms, or by investing further human effort in trawling through lists of even more unlikely potential matches, but, for the purposes of a time-limited case study, the effort required to go through lists of hundreds of unlikely matches in order to find a single new match would not have been justified.

To conclude this section, we provide some statistics on the start and end dates of the true positives that we have matched. Earlier, we remarked that these dates were too unreliable to base automated procedures on, but now that we have compiled a sample of 118 actual matches, we can analyse how informative their registered start and end dates could be. A statistical summary is shown in the table below:

	START DATE	END DATE
Exact match	64	1
Between 1 and 7 days	5	0
More than a week, no more than a month	10	0
More than a month, no more than a year	12	2
More than a year	8	2
One dummy date	14	25
Two dummy dates	5	88

We can conclude that, while automated procedures would discard some true positives due to the dates being considerably different, there are many cases in which the dates can serve as useful verification tools. Start dates, in particular, are well populated in the databases, and more than half of the identified matched corporation do, in fact, have exactly the same start dates in both databases. Some others have start dates that are not too far apart, though there are quite a few cases where the start dates are more than a year apart. When it comes to end dates, nearly all cases have a dummy date in at least one of the databases, and a large majority in both. This is partly due to the fact that many corporations are still active and therefore no end date could be registered, but in some cases this is due to the fact that the end date is unknown. In just one case, the matched corporations had the same effective end date in both databases. At the other extreme, in one case the two end dates were over 30 years apart.

## Matching Securities

In order to match securities, we use the hierarchical nature of the data in the databases. Each security belongs to an issuer, and we can therefore use the corporation matching data we discovered above as a stepping stone towards security matching. In short, we will only attempt to match two securities if we know they belong to the same corporation. Additionally, we will only match securities of the same type – for example, a stock will only be matched to stocks of the same type, a bond to bonds of the same type, etc.

By doing this, we dramatically reduce the use of our resources. First of all, our automated methods will need a lot less computation time to produce potential matches, and, second, the human experts that will need to inspect these matches will be given fewer cases to verify. Concretely, in the period between 1 January 1890 and 31 December 1906, the SCOB database contains data on 3489 securities, and the DFIH database contains data on 1620 securities. Without our hierarchical approach, we would need to make over 5 million comparisons, most of them entirely spurious. However, after filtering for the corporations we matched in the previous phase, and then separating stocks from bonds, we found that only 24 of the 118 matched corporations had stocks traded at both stock exchanges, and only 17 had bonds traded at both stock exchanges. Some of the corporations had multiple stocks and bonds traded at the two stock exchanges, so the total number of comparisons we needed to make turned out to be 74 for stocks and 88 for bonds, several orders of magnitude fewer than the 5 million comparisons needed if we approached this task blindly.

## Security Name Matching

As with corporations, our first attempt was limited to matching security names. In both databases, we ran a query that listed all corporation IDs, stock IDs, stock names, as well as start and end dates, of all securities (note that both stocks and bonds have “stock” IDs and names in the two databases) listed at the respective stock exchanges at some point between 1 January 1890 and 31 December 1906, that belonged to the corporations that have been matched in phase 1 of the case study. This produced 642 stock names in SCOB and 386 stock names in DFIH. Naturally, as discussed above, we do not need to compare all 642 stock names from SCOB to all 386 stock names from DFIH, but only those pairs that belong to the same corporation, further reducing the required effort.

Nevertheless, the results of this first experiment were far from spectacular. Using the normalised Levenshtein distance with a threshold of 0.2, we found no results at all. Increasing the threshold to 0.3, we obtained the following results (note that ID and NAME now refer to stock ID and stock name, respectively):

SCOB.ID	SCOB.NAME	DFIH.ID	DFIH.NAME	DISTANCE
27689	Banlieue de Reims et extensions (Chemins de fer) (action de capital)	6723	Banlieue de reims et extensions (Chemins de fer de la)	0.221

712	Vireux Molhain (Forges) (?1893)	106585	VIREUX MOLHAIN (forges de)	0.226
29121	Banlieue de Reims et extensions (Chemins de fer) (action de jouissance)	6723	Banlieue de reims et extensions (Chemins de fer de la)	0.254
1473	Wagons-Lits (Cie Internationale des) (act.priv.) (?1902)	3410	Wagons-lits (Cie internationale des) act.500fr.t.p.	0.268
26508	Metropolitain de Paris (estampille)	17241	Métropolitain de Paris, actions estampillées	0.273
558	Escombrera-Bleyberg (Comp Franc des Mines et Usines d') (1 a 40.000)	2881	Escombrera-Bleyberg (Cie Françse des Mines et Usines d'), act. 350 fr., t. p.	0.286

When we increased the threshold to 0.4, we obtained another 5 potential matches. In other words, of all possible pairs of stock names, all but 11 of them had a distance of more than 40%. This clearly demonstrated that we could not rely on stock names at all for the stock matching task. The stock names, as currently stored in the databases, often include information that should probably be separated over multiple attributes (such as type, year, value, or even loose comments). Furthermore, just as with corporations, the start and end dates differed widely, and dummy dates were even more prevalent. We therefore decided to focus our efforts on matching stocks using stock prices and dividends.

In the next few sections, we discuss matching stocks and bonds separately, but the methods were largely the same.

### Stock Price Matching

In this experiment, we limit ourselves to stocks belonging to the corporations matched in the previous phase of the case study. As a first step, we exported all the prices of these stocks, in the period between 1 January 1890 and 31 December 1906, from the two stock exchanges. We observed that the SCOB database in most cases contained one price per month for each stock, while the DFIH database was less consistent, with sometimes multiple prices per month, and sometimes several months without a price. As a result, we decided to try to match prices month per month, taking the average of all prices within a month in cases where there were more than one. In future, this approach could be optimised by taking the nearest price, or by comparing only prices within a certain number of days from each other. Additionally, this case study was facilitated by the fact that the French and Belgian franc had exactly the same value in the analysed period. In other cases, we may have to rely on currency exchange rates, which may not always be present in the database.

A second observation was that prices could be very different in scale (from a few francs to thousands of francs). We therefore concluded that the price matching should not be done based on absolute differences between the prices, but on relative differences, or ratios (e.g., the difference between 9 and 10 is larger than the difference between 99 and 100). Concretely, in each month in which we found a price in both databases, we divided the smaller average price with the larger average price to compute the ratio for that month. After doing this for all months, we computed the average of all these monthly ratios as the final similarity score for the given pair of stocks.

Finally, a third observation was that for some pairs of stocks there were only very few months (sometimes just one month) in which both stocks had a price registered at their respective stock



exchange. As a result, we decided to focus only on pairs of stocks for which we had at least 12 matched months (this is a flexible parameter that can be changed in future experiments).

With a similarity threshold set at 0.9, we discovered 29 potential matches. After human inspection, it turned out 25 of those were true positives, giving a true positive rate of  $25/29 = 86.21\%$ . With a threshold set at 0.97, the true positive rate was  $19/20 = 95\%$ . Lowering the similarity threshold to 0.8 produced no new matches. We therefore concluded that 0.9 was a sensible threshold producing very satisfactory results.

It is interesting to note that for some stocks, we found multiple correct matches. The reason for this is that the SCOB database uses different IDs for the same stock traded at different markets. Furthermore, the SCOB database contains information not only of the Brussels Stock Exchange, but also of the Antwerp Stock Exchange that existed at the time. As a result, some stock IDs in DFIH were matched with up to three different stock IDs in SCOB. If an end user wishes to focus only on a particular market, an appropriate filter can be performed.

### Stock Dividend Matching

In a further experiment, we attempted to match stocks based on dividends registered in the two databases. However, we observed that much of the dividend information was missing, and our experiments ultimately discovered only a subset of the matches already discovered above. We conclude that the dividend information can be a useful tool for verifying potential matches, rather than identifying them. However, while the results were very limited within our case study on a small subset of the data, the developed methods may well prove useful in a future iteration on more complete and up-to-date datasets.

### Bond Price Matching

Our final experiment was also the most difficult one. The bond market is especially challenging, as many corporations and, even more so, national and local governments, typically issue many very similar bonds within overlapping periods. As such, these bonds have similar terms and, in our context more importantly, are traded at similar prices. For example, if ten bonds of a company are traded at one stock exchange, and ten at another, all with similar prices, our methods would identify 100 potential matches, of which at least 90% would be false positives.

To alleviate this issue, we limit this case study to corporations, and omit government bonds from our analysis. The methodology we used was the same as used for stocks above: we computed a similarity measure between pairs of bonds based on average monthly prices, and then reported those pairs that had a similarity higher than 0.9 and at least 12 matched months. This gave us 28 pairs of bonds that represented potential matches. After verification performed by a human expert, it turned out that 16 of these were correct matches, giving us a true positive rate of  $16/28 = 57.14\%$ . This was considerably lower than the results we achieved in corporation matching and stock matching, but was nevertheless expected in the circumstances. While our experiment showed that we were able to identify some matches, there is clearly scope for improvement. In future iterations, we intend to investigate if other information, that currently may not be consistently present in the databases, may help the matching task.



If we examine the results in more detail, we see that two corporations in particular hampered our algorithms. One bond issued by *Compagnie du Chemin de Fer du Nord* that was traded at the Brussels Stock Exchange was matched with five different bonds traded at the Paris Stock Exchange. This, naturally, led to four false positives, as only one of the five potential matches was correct. An interesting observation was that the true positive had a similarity score of 0.99, while the other four all scored 0.97 or lower, so ranking them on score and suppressing further potential matches if a match has already been found would lower the effort required for human verification. However, as we have seen when matching stocks, a security at one stock exchange can sometimes have multiple true matches at another stock exchange, so suppressing further recommendations may not always be desirable.

Another corporation that negatively affected our performance was *Compagnie de Chemins de Fer Départementaux*. This corporation had three similar bonds traded in Brussels and the same three in Paris. Since their prices were similar, our algorithms produced nine potential matches, of which three were correct and six wrong. Here, in two of the three cases, the correct match had a higher score than the two false positives.

Taking these two corporations out of the sample, of the remaining 14 potential matches, 12 proved correct, with the true positive rate of  $12/14 = 85.71\%$ . The only remaining false positives were two cases where we matched a bond from the Brussels Stock Exchange to two different bonds at the Paris Stock Exchange. In both cases, one of the two matches was correct, and in both cases, the match with the higher score was correct.

## Case 2: Developing a Collaborative Environment

### Context

Historical firm level data exists in a wide range of formats, digital or otherwise, within the EURHISFIRM project itself and beyond. Some research teams within the EURHISFIRM project have been working with their own collections of research material and building their own databases for a long time. While EURHISFIRM starts developing its services around a common core of data, these source databases will continue to function for some years for storing and editing data beyond the scope of this common core. Consequently any proposition willing to further the goals of the project and to help the teams working together more effectively needs to function with the many different databases, tools and organisational structures that are in use now and for the foreseeable future. We have to assume that the source databases will always keep changing, so the proposition needs to be able to evolve without being rendered obsolete by the underlying changes in the source data.

Therefore, the need is for an autonomous, self-contained environment, distinct from the source databases, which would allow the research teams to collaboratively import, edit and use data from all source databases. The goal is to facilitate the process of matching entities of all types between all sources, and ultimately visualise and export the results of this collaborative work for their own needs.

We propose to use Wikibase to develop this new EURHISFIRM collaborative environment.



Wikibase would not provide by itself the tool to *find* the matches between entities in separate databases but would allow to *register* and share those matches in a way that can be used by others. In this way, many matching techniques and processes could be thought out, experimented and used independently by any parties. Those would be run outside of Wikibase while their findings would be automatically registered by a program (bot) and then verified by humans, all in a centralised and open database on the web: the EURHISFIRM Wikibase platform. In other words, Wikibase is not the database to rule them all, but the database to link them all.

This case study will demonstrate how Wikibase can address those established needs and objectives, starting with a small sample from the SCOB and DFIH teams.

All the source code of this work – past, present and future – is accessible on the repositories at this address: <https://gitlab.huma-num.fr/eurhisfirm>.

The sandbox instance of Wikibase used for this work is accessible at this address: [data.eurhisfirm.eu](https://data.eurhisfirm.eu). Please note that this is currently used as an experimental and development test bed and is not intended for users. Anything on this website, such as pages and data used to write this case study, will change and can be wiped at any time. Consequently, there's currently no need to restrict the contribution from users. This can and will be changed, according to the defined EURHISFIRM governance and accreditation system.

## Wikibase

[Wikibase](#) is the main piece of software powering the [Wikidata](#) project, a relatively recent sister project of [Wikipedia](#). Wikidata is to structured data what Wikipedia is to raw, unstructured content.

Every Wikipedia *article* has an equivalent as a Wikidata *item*. While the former is only understandable by a human, the latter can be read and interpreted by a machine. Indeed, several virtual assistants such as Amazon Alexa or Google Assistant already use Wikidata to answer questions asked by real people.





## Napoleon

From Wikipedia, the free encyclopedia

*This article is about Napoleon I. For other uses, see [Napoleon \(disambiguation\)](#).  
 "Napoleon Bonaparte" redirects here. For other uses, see [Napoleon Bonaparte \(disambiguation\)](#).*

**Napoleon Bonaparte** (born **Napoleone di Buonaparte** (Italian: [napoleˈoːne di ˈbwɔnaˈparte]; French: Napoléon [napolɛ̃ ˈbɔnapaʁt]<sup>[1]</sup> 15 August 1769 – 5 May 1821) was a French statesman and military leader who rose to prominence during the French Revolution and led several successful campaigns during the French Revolutionary Wars. He was Emperor of the French as **Napoleon I** from 1804 until 1814 and again briefly in 1815 during the Hundred Days. Napoleon dominated European and global affairs for more than a decade while leading France against a series of coalitions in the Napoleonic Wars. He won most of these wars and the vast majority of his battles, building a large empire that ruled over much of continental Europe before its final collapse in 1815. He is considered one of the greatest commanders in history, and his wars and campaigns are studied at military schools worldwide. Napoleon's political and cultural legacy has endured as one of the most celebrated and controversial leaders in human history.<sup>[2][3]</sup>

He was born in Corsica to a relatively modest Italian family from minor nobility. He was serving as an artillery officer in the French army when the French Revolution erupted in 1789. He rapidly rose through the ranks of the military, seizing the new opportunities presented by the Revolution and becoming a general at age 24. The French Directory eventually gave him command of the Army of Italy after he suppressed the 13 Vendémiaire revolt against the government from royalist insurgents. At age 26, he began his first military campaign against the Austrians and the Italian monarchs aligned with the Habsburgs—winning virtually every battle, conquering the Italian Peninsula in a year while establishing "sister republics"



instance of

human

edit

1 reference

+ add value

image

Jacques-Louis David - The Emperor Napoleon in His Study at the Tuileries - Google Art Project 2.jpg  
3,076 × 5,117; 6.77 MB

0 references

+ add reference

+ add value

sex or gender

male

edit

2 references

*Napoleon on Wikipedia and Wikidata (click to open)*

Wikibase being open source, it can be installed and used by anyone. Actually, there are many existing instances of Wikibase, while Wikidata remains by far the largest one.

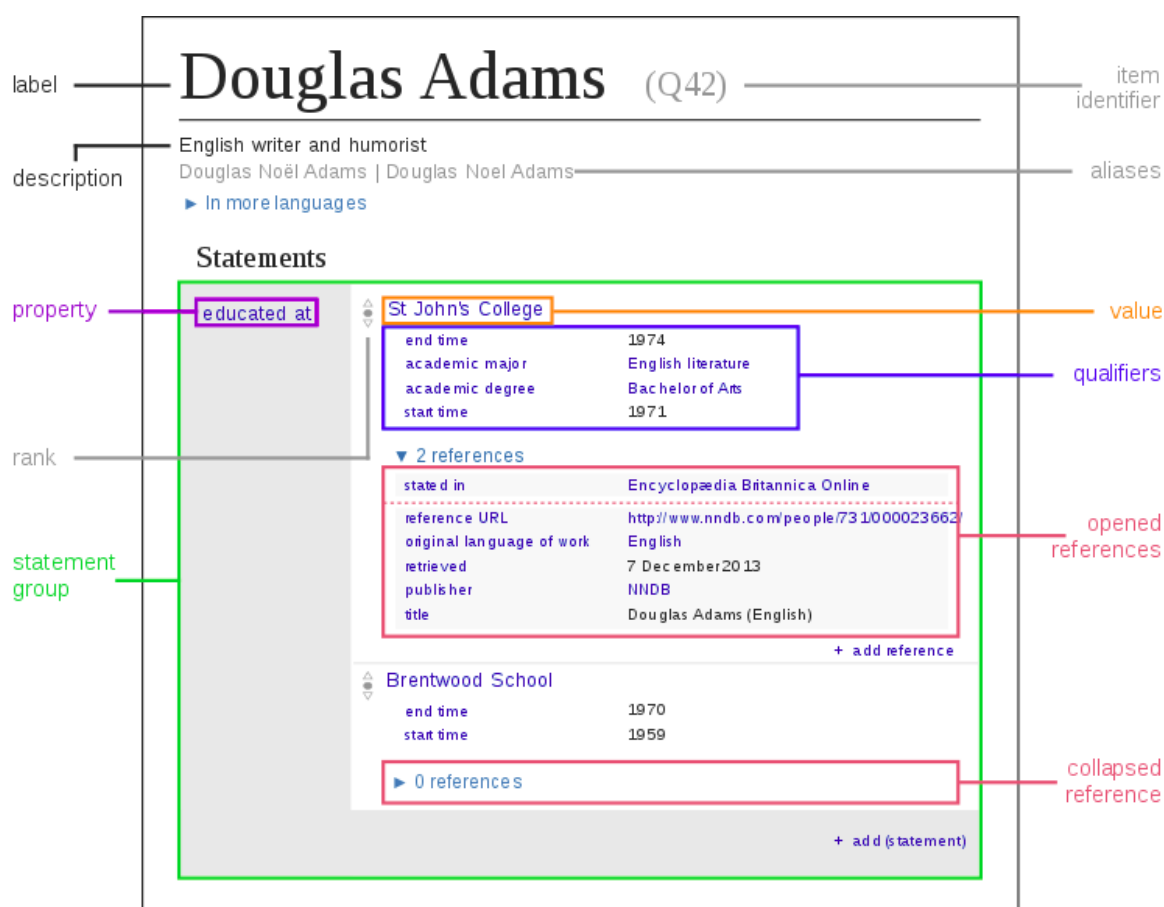
A Wikibase instance consists mainly of pages known as *items* and *properties*. They are the main building blocks of a Wikibase instance. We call them *entities*. In essence they are respectively the data and the relations that structure a Wikibase database.

All of the *entities* on an instance have a unique identifier starting with a Q (*items*) and a P (*properties*), as well as a *label*, a *description* and any number of *aliases*, all of them in as many languages as needed.

As an example, here's the item pertaining to the late British author Douglas Adams, [Q42 in Wikidata](#):







Entity characteristics are described by *statements* which consist of a *property* and a *value*.

*Qualifiers* consist themselves of a *property* and a *value*. Applied to statements, they allow them to be expanded on, annotated, or contextualized. Finally, *references* allow to indicate where the data comes from and add sources.

Which language the labels are displayed on can be configured by the user, as long as the translations have been added.

### Experiment with DFIH and SCOB data and first results

This first case study of Wikibase shows the current state of the team's work, i.e., importing, editing and using data on a dedicated instance of Wikibase for EURHISFIRM. The scope of the case study is deliberately limited as to highlight only the main features of Wikibase used to address the simple needs and objectives listed above.

### Data

The EURHISFIRM data we are currently working with is a sample from the SCOB and DFIH databases, containing only information about:

- Companies: ID, names, legal form, location and relevant dates;

- Persons: ID, name, gender, position held and relevant dates.

### Steps

Once exported from their source database, those samples consist of tabular files in the CSV format, which are then handled by scripts of our own. The case study consists of several steps:

- Import entities from the SCOB and DFIH databases;
- “Enrich” data, i.e. edit information via the Wikibase interface;
- Match items from SCOB and DFIH;
- Merge two items into one Wikibase item;
- Query Wikibase, e.g., list and display companies on a map using their location.

### Import entities

There are many different ways to add and import data in Wikibase, manually or automatically. Tools with user interfaces such as [OpenRefine](#), [QuickStatements](#) or [Mix’n’match](#) will also be explored as part of our ongoing work but were not used for this case study.

For the import we used [WikidataIntegrator](#), a powerful open source Python program with a large Wikidata/Wikibase user and developer community. Using this library, we wrote Python scripts adapted for the SCOB and DFIH databases in order to correctly represent the source data from its original format (tables, columns, line...) to the Wikibase model (items, properties, values...).



## Reading and editing an item

To show an example, we used the company corresponding to the ID number 1040 in the DFIH database, named “Banque Nationale pour le commerce et l'Industrie”.

We can see below that the source data has been successfully imported and converted to the Wikibase data model we explained above. Just after this initial import, we can already start to make use of the features of Wikibase to enrich the data via the interface.

For example, we added English labels when missing in the source data, which we used when taking the screen captures displayed in this report.

**EURHISFIRM**  
 Long-term data for Europe

[Main page](#)  
[Recent changes](#)  
[Random page](#)  
[Help about MediaWiki](#)

**Tools**  
[What links here](#)  
[Related changes](#)  
[Special pages](#)  
[Printable version](#)  
[Permanent link](#)  
[Page information](#)  
[Concept URI](#)

In other languages  
[Add links](#)

Item Discussion

## National Bank for Trade and Industry (Q162)

No description defined

▼ [In more languages](#) [Configure](#)

Language	Label	Description	Also known as
English	National Bank for Trade and Industry	No description defined	
French	Banque nationale pour le commerce et l'industrie	No description defined	

### Statements

<a href="#">corporation name</a>	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> </div> <div>             Banque Nationale pour le commerce et l'Industrie             <span style="float: right;"><a href="#">edit</a></span> </div> </div> <div style="margin-top: 5px;"> <div style="display: flex; justify-content: space-between;"> <div><a href="#">start date</a></div> <div>18 April 1932</div> </div> <div style="display: flex; justify-content: space-between;"> <div><a href="#">end date</a></div> <div>31 December 3999</div> </div> </div> <div style="margin-top: 5px;"> <a href="#">1 reference</a> </div>
----------------------------------	--

Any of those statements can be edited, modified, corrected and enriched through the user interface with just one click. References to the source material can and should be given for any statement. A versioning history for each Wikibase entity, similar to the one seen on Wikipedia, allow to keep track of all changes and even revert them if needed.

start date	<div> <div>1 January 1000 <i>Gregorian</i></div> <div>edit</div> </div> <div> <div>1 reference</div> <div>+ add value</div> </div>
end date	<div> <div>31 December 3999</div> <div>edit</div> </div> <div> <div>1 reference</div> <div>+ add value</div> </div>
legal form	<div> <div>Public limited-liability company</div> <div>edit</div> </div> <div> <div>start date</div> <div>18 April 1932</div> </div> <div> <div>end date</div> <div>31 December 1945</div> </div> <div> <div>comments</div> <div>Société Anonyme Française constituée le 18 avril 1932. Nationalisée à compter du 1er janvier 1946, en vertu de la loi du 2 décembre 1945.</div> </div> <div> <div>1 reference</div> </div>
	<div> <div>Q149</div> <div>edit</div> </div> <div> <div>start date</div> <div>1 January 1946</div> </div> <div> <div>end date</div> <div>31 December 3999</div> </div> <div> <div>comments</div> <div>Société Anonyme Française constituée le 18 avril 1932. Nationalisée à compter du 1er janvier 1946, en vertu de la loi du 2 décembre 1945.</div> </div> <div> <div>1 reference</div> </div> <div> <div>+ add value</div> </div>
instance of	<div> <div>Company</div> <div>edit</div> </div> <div> <div>1 reference</div> <div>+ add value</div> </div>
DFIH corporation ID	<div> <div>1040</div> <div>edit</div> </div> <div> <div>1 reference</div> <div>+ add value</div> </div>
location	<div> <div>Q134</div> <div>edit</div> </div> <div> <div>location type</div> <div>Q142</div> </div> <div> <div>in yearbook from</div> <div>1933</div> </div> <div> <div>in yearbook to</div> <div>1953</div> </div> <div> <div>1 reference</div> </div>
	<div> <div>Q135</div> <div>edit</div> </div> <div> <div>location type</div> <div>Q144</div> </div> <div> <div>in yearbook from</div> <div>1945</div> </div> <div> <div>in yearbook to</div> <div>1945</div> </div> <div> <div>1 reference</div> </div>

## Matching

Matching items such as companies or persons from separate databases can be done in using many different methodologies, some of which have been explored in the first case study described in this report.

The interest of this second case study is not in the matching methodology as such, but in the ways we can facilitate multiple groups of people working with different matching methodologies to work together.

For this example, we found multiple matches of companies from SCOB and DFIH, using a program with a very simple methodology based on finding companies with exactly the same names. Once manually verified that all characteristics were also equal, we could confirm for this demonstration that at least one company appears in the SCOB and DFIH DBs: “Caisse départementale de la Mayenne”.

The actual database IDs and complete names from SCOB and DFIH were:

- SCOB Corp ID 20430: “Caisse départementale de la Mayenne (Commandite, Raison sociale E. Chanteau et Cie - primitivement F.Piquet et Cie, puis Chnteau, Ve Veillard et Cie) (SOURCE : COURTOIS 1863 128) (SOURCE: COURTOIS 1878 201)”
- DFIH Corp ID 1000: “Caisse départementale de la Mayenne (Raison sociale E. Chanteau et Cie)”

This company was part of the sample imported in the Wikibase for experiment. We can retrieve both items using their source DB IDs. They were imported separately and as such had two separate item ID on Wikibase: Q205 for the item from DFIH and Q70 for the one from SCOB.

## Merging

Wikibase provides a built-in tool to merge two items, mostly for when there are duplicates. Its interface is simple and intuitive, requiring no previous knowledge.

In this case we decided to merge Q205 (“ID to merge from”) into Q70 (“ID to merge to”).

## Merge two Items

### Merge two Items

If you merge two Items, all labels, descriptions, aliases, sitelinks and statements will be moved from one Item to the other.

The ID to merge from

The ID to merge to

Merge Items

The result is exactly as desired: The page of the Q205 item now redirects to the Q70 item and all information from the former was added to the latter. If all information was exactly the same, the two items would be merged while still containing the references from before the merge.

corporation name	<div> <div></div> <div> Caisse départementale de la Mayenne (Commandite, Raison sociale E. Chanteau et Cie - primitivement F.Piquet et Cie, puis Chnteau, Ve Veillard et Cie) (SOURCE : COURTOIS 1863 128) (SOURCE: COURTOIS 1878 201) </div> <div>edit</div> </div> <div> <div>start date</div> <div>25 May 1847 <i>Gregorian</i></div> </div> <div> <div>end date</div> <div>31 December 3999</div> </div> <div> <div>▼ 1 reference</div> <div> <div>source database</div> <div>SCOB database</div> </div> <div> <div>retrieved</div> <div>25 March 2020</div> </div> <div>+ add reference</div> </div>
	<div> <div></div> <div> Caisse départementale de la Mayenne (Raison sociale E. Chanteau et Cie) </div> <div>edit</div> </div> <div> <div>start date</div> <div>25 May 1847 <i>Gregorian</i></div> </div> <div> <div>end date</div> <div>31 December 3999</div> </div> <div> <div>▼ 1 reference</div> <div> <div>source database</div> <div>DFIH database</div> </div> <div> <div>retrieved</div> <div>25 March 2020</div> </div> <div> <div>source</div> <div> COURTOIS 1863 128 **  COURTOIS 1863 128 **  COURTOIS 1878 201 </div> </div> <div>+ add reference</div> </div> <div>+ add value</div>

Here, the “corporation name” property, which was used to model the imported company names from SCOB and DFIH, now contains two values, one for each appearance in the source databases. We can track the origin of each with the “reference” information stated for each value, with an additional

“retrieved” date (stating when this information was imported in Wikibase) and optional “source” info (comments stated in the database).

start date	 30 December 1000 <i>Gregorian</i>  edit
	▼ 1 reference
	source database SCOB database
	retrieved 25 March 2020
	+ add reference
	 1 January 1000 <i>Gregorian</i>  edit
	▼ 1 reference
	source database DFIH database
	retrieved 25 March 2020
	+ add reference
+ add value	

end date	 31 December 3999  edit
	▼ 2 references
	source database SCOB database
	retrieved 25 March 2020
	+ add reference
	source database DFIH database
	retrieved 25 March 2020
	+ add reference
+ add value	

Above we can see two statements using “start date” and “end date” properties we created to model the information about times of creation and dissolution of a company, when available. Current practices in the DFIH and SCOB databases is to use the years 1000 and 3999 when information is unavailable, so we can consider those statements as null, even though the start dates of the merge items do not exactly match (different day and month). In fact, we could even remove those statements without missing any actual bit of information.

We can also see the trace of that merge in the history of both items, first in the Q70 item:

- (cur | prev)  18:25, 29 March 2020 Johan (talk | contribs) . . (154 bytes) (-7,818) . . (Merged Item into Q70) (undo) (Tag: Replaced) (restore)
- (cur | prev)  18:25, 29 March 2020 Johan (talk | contribs) . . (55 bytes) (-99) . . (Redirected to Q70) (undo) (Tag: New redirect)
- (cur | prev)  08:22, 25 March 2020 WikibaseAdmin (talk | contribs) . . (7,972 bytes) (+7,972) . . (Created a new Item) (restore)

and in the Q205 item:

- (cur | prev) ☐ 18:25, 29 March 2020 Johan (talk | contribs) . . (15,078 bytes) (+6,843) . . (Merged Item from Q205) (undo) (restore)
- (cur | prev) ☐ 08:19, 25 March 2020 WikibaseAdmin (talk | contribs) . . (8,235 bytes) (+8,235) . . (Created a new Item) (restore)

A “restore” option is given to revert the merge, for example if it is decided that the merge was in fact wrong.





## Exploitation

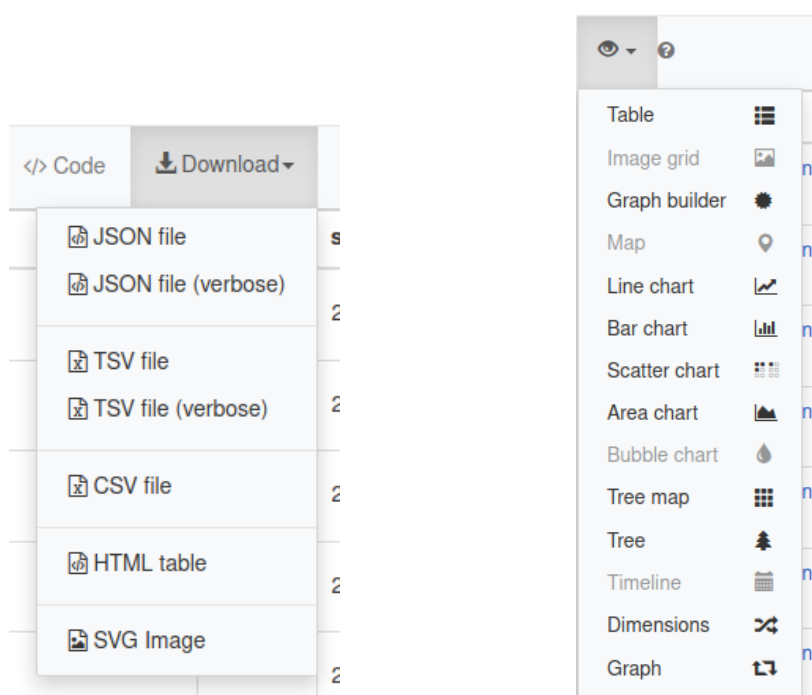
Beyond the tooling provided by Wikibase to integrate (import) and enrich (edit) the data, there are many interesting prospects as part of the exploitation of the data: validation, visualisation, web publication, conversion, reuse, export, etc.

Wikibase data can be queried, i.e., retrieved from specific criteria, using a special tool, the “Query Service”, and the SPARQL language which can be compared to the standard SQL language.

For example, we can retrieve the list of all the companies in the Wikibase instance, with their label as well as their SCOB and/or DFIH ID, with just a few lines of SPARQL code:

company	companyLabel	dfihld	scobld
<a href="http://wikibase.svc/entity/Q100">Q &lt;http://wikibase.svc/entity/Q100&gt;</a>	Banque hypothécaire et d'escompte de Bavière (Baierische Hypotheken und Wechsel Bank) (S.A.) (SOURCE: COURTOIS 1863 158)		20491
<a href="http://wikibase.svc/entity/Q101">Q &lt;http://wikibase.svc/entity/Q101&gt;</a>	Banque de Leipzig (Leipziger Bank) (S.A.) (SOURCE: COURTOIS 1863 160)		20495
<a href="http://wikibase.svc/entity/Q102">Q &lt;http://wikibase.svc/entity/Q102&gt;</a>	Banque Nationale pour le commerce et l'Industrie (SOURCE : ANNUAIRE DESFOSSES 1956 19)		20498
<a href="http://wikibase.svc/entity/Q103">Q &lt;http://wikibase.svc/entity/Q103&gt;</a>	Banque de Francfort (Frankfurter Bank) (S.A.) (SOURCE: COURTOIS 1863 171)		20504
<a href="http://wikibase.svc/entity/Q104">Q &lt;http://wikibase.svc/entity/Q104&gt;</a>	Banque de Hombourg (Landgräfllich Hessische concessionirte Lands-Bank in Homburg vor der Höhe) (S.A.) (SOURCE: COURTOIS 1863 173) (date exacte inconnu)		20506
<a href="http://wikibase.svc/entity/Q105">Q &lt;http://wikibase.svc/entity/Q105&gt;</a>	Banque de l'Allemagne méridionale (Bank für Süd-Deutschland) (S.A. allem) (autorisée par D.G.D.) (SOURCE: COURTOIS 1863 174)		20507

We can also directly download the data in several formats or visualise it in different modes, when suited:



As part of this case study we will explore one of those visualisations: geographic mapping. But first we need actual geo data. Since the SCOB and DFIH databases contain information about the locations of companies, we can use and model them in order to create maps.

For example, the company “Comptoir Central” (SCOB Corp ID 20389) is said to have been located at “31 rue Mogador, Paris, France” in 1904.

SCOB corporation ID	20389	<a href="#">edit</a>
	<a href="#">1 reference</a>	
	<a href="#">+ add value</a>	

location	31 rue Mogador, Paris, France	<a href="#">edit</a>
	start date	31 December 1903 <i>Gregorian</i>
	end date	31 December 1904 <i>Gregorian</i>
	<a href="#">1 reference</a>	
	source database	SCOB database
	retrieved	25 March 2020
	<a href="#">+ add reference</a>	
	<a href="#">+ add value</a>	

As you can see in the company statement above, we imported this information in Wikibase by actually creating a dedicated item (i.e., a page) for this location. This allows us to store all the address details about this location without cluttering up the company item:

address	rue Mogador 31	<a href="#">edit</a>
	<a href="#">1 reference</a>	
	<a href="#">+ add value</a>	

city	Paris	<a href="#">edit</a>
	<a href="#">1 reference</a>	
	<a href="#">+ add value</a>	


  

country	France	<a href="#">edit</a>
	<a href="#">1 reference</a>	
	<a href="#">+ add value</a>	

Also, if we actually found out that this street address information is only relevant to 1904, for example if the street name changed later on, we could also represent this information.

For now, we considered that this address is still relevant to this day, and we added the geographic coordinates to the address using a [geocoder](#):

coordinate location

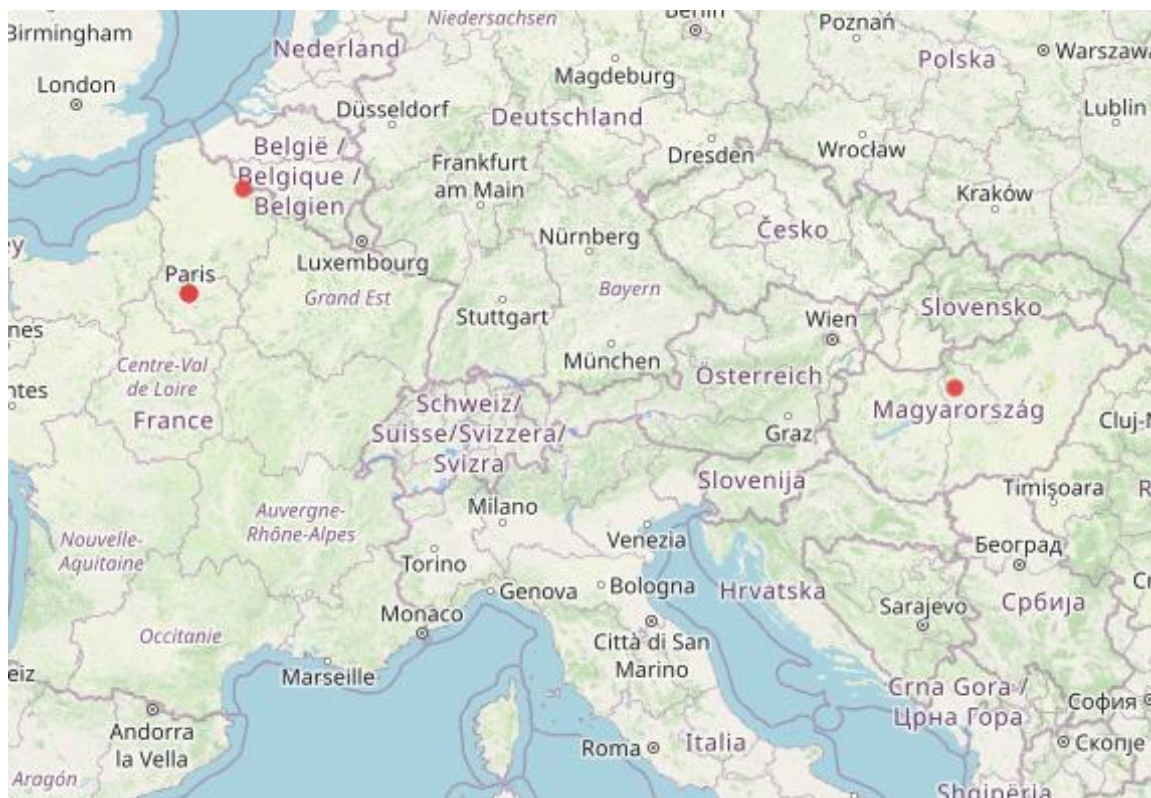
 48°52'32.765"N, 2°19'51.996"E  
 ▼ 0 references

This geocoding (conversion from postal address to geo coordinates) could be done automatically, and added to all address items in the instance. For this experiment we added the information manually to just a few items.

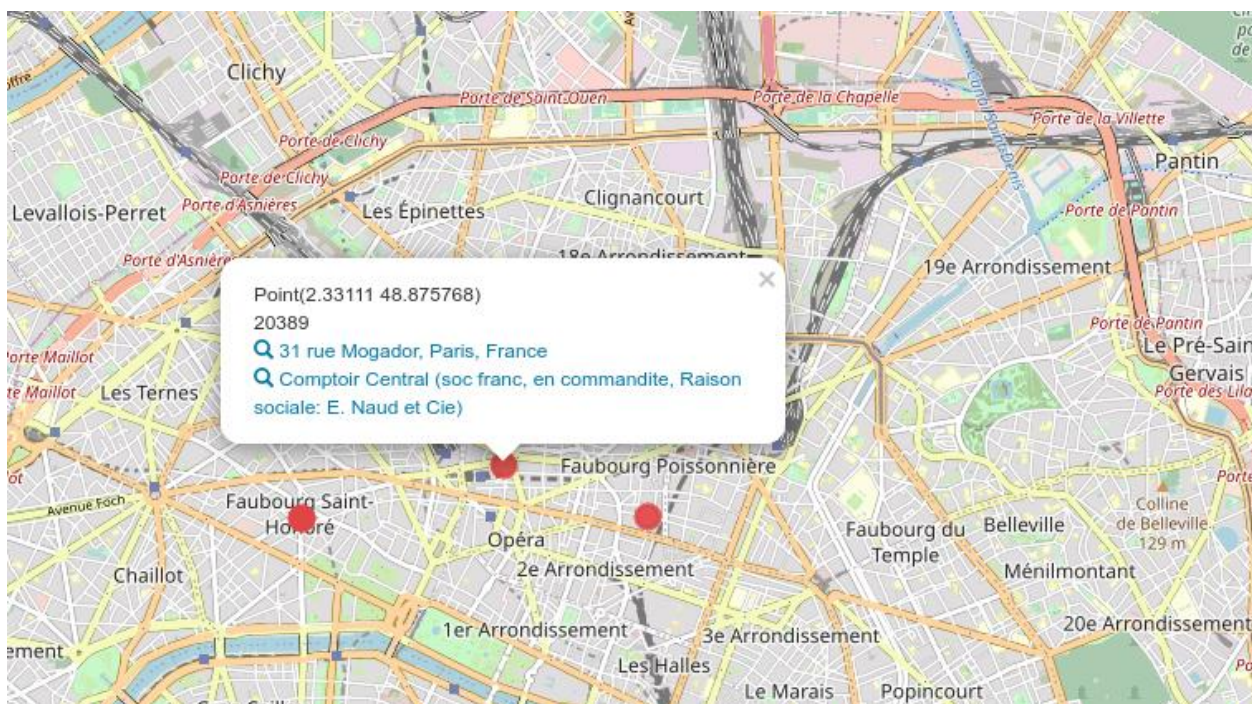
Since Wikibase knows how to interpret geographic coordinates we can query the same list as before, this time asking the coordinate location when available:

company	companyLabel	location	locationLabel	geoCoordinates	dfihld	scobld
 <a href="http://wikibase.svc/entity/Q158">&lt;http://wikibase.svc/entity/Q158&gt;</a>	Banque Louis Dupont et Cie	 <a href="http://wikibase.svc/entity/Q131">&lt;http://wikibase.svc/entity/Q131&gt;</a>	68 rue du quesnoy, Valenciennes, Nord, France	Point(3.52646 50.356906)	1034	
 <a href="http://wikibase.svc/entity/Q175">&lt;http://wikibase.svc/entity/Q175&gt;</a>	BANQUE COMMERCIALE HONGROISE DE PEST	 <a href="http://wikibase.svc/entity/Q140">&lt;http://wikibase.svc/entity/Q140&gt;</a>	Rue bergère, 14, Paris, Paris, France	Point(2.346151 48.87232)	1060	
 <a href="http://wikibase.svc/entity/Q46">&lt;http://wikibase.svc/entity/Q46&gt;</a>	Comptoir Central (soc franc, en commandite, Raison sociale: E. Naud et Cie)	 <a href="http://wikibase.svc/entity/Q16">&lt;http://wikibase.svc/entity/Q16&gt;</a>	31 rue Mogador, Paris, France	Point(2.33111 48.875768)		20389
 <a href="http://wikibase.svc/entity/Q158">&lt;http://wikibase.svc/entity/Q158&gt;</a>	Banque Louis Dupont et Cie	 <a href="http://wikibase.svc/entity/Q130">&lt;http://wikibase.svc/entity/Q130&gt;</a>	26 avenue franklin-roosevelt, Paris, France	Point(2.310215 48.872196)	1034	
 <a href="http://wikibase.svc/entity/Q175">&lt;http://wikibase.svc/entity/Q175&gt;</a>	BANQUE COMMERCIALE HONGROISE DE PEST	 <a href="http://wikibase.svc/entity/Q141">&lt;http://wikibase.svc/entity/Q141&gt;</a>	Budapest, Hongrie	Point(19.0544 47.4991)	1060	

And here's the same information when we choose the map visualisation:



Zooming in on Paris:





## Conclusion

In this report, we described two case studies that deal with the challenges of matching data originating in the individual databases belonging to partners within the EURHISFIRM consortium.

The first case study, performed by the Antwerp team, investigated a number of data matching techniques and their respective usefulness in the task of matching company and security data from the SCOB and DFIH databases. A number of techniques were tested and their performance evaluated using the false positive rate, while simultaneously keeping an eye on the human effort need to supervise the automatic algorithms and verify their output. When matching companies, we concluded that the most reliable piece of information was the company name itself, while the best performance was achieved with the normalised edit distance metric. When matching securities, on the other hand, security names proved less useful, and more accurate results could be obtained relying on security prices and dividends, using specially designed similarity measures. On top of the challenges encountered when actually designing and implementing data matching algorithms, one of the main difficulties we encountered was that of data quality. The data quality issues ranged from simple inconsistencies in spelling to sometimes using the “name” attribute in the databases to store not just the name, but also other information or loose comments. When it came to numerical data, such as security prices or dividends, we encountered cases of erroneous information, missing data, and even duplicates. Some of these issues can be overcome by using approximate data matching techniques, but to achieve optimal results, data should be as clean and as correct as possible.

The second case study, performed by the Paris team, investigated the potential of using Wikibase to develop a collaborative environment that could lead to a user-friendly way of using the integrated EURHISFIRM database in the future. The task of data matching could clearly benefit from such an environment, too. Wikibase is relatively easy to use without much prior knowledge, certainly compared to specialised database systems currently in use by some EURHISFIRM partners. As such, Wikibase cannot directly be used to *identify* matches in the data, but it can be very easily used to *register* such matches. In fact, it literally takes just a few clicks to merge two Wikibase items into one item.

Both case studies resulted in useful insights taking the EURHISFIRM project forward. In the next step, we will attempt to link the data available within the EURHISFIRM project to data available from external sources. This will bring additional challenges, but both the data matching techniques discussed in Case 1, and the collaborative environment developed in Case 2, will undoubtedly prove to be valuable building blocks in the coming tasks.

## References

Hautcoeur, P. C. and A. Riva (2012). The Paris financial market in the nineteenth century: complementarities and competition in microstructures. *Economic History Review*, 65 (4), 1326-1353



Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In Soviet physics doklady (Vol. 10, No. 8, pp. 707-710)

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage

