Long-term data for Europe

EURHISFIRM

M6.2: Data Connecting Case Study

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

http://www.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)

APPROVED IN 2020 BY:

Jan ANNAERT (UNIVERSITEIT ANTWERPEN)
Wolfgang KÖNIG (JOHANN WOLFGANG GOETHE-UNIVERSITÄT FRANKFURT AM MAIN)
Angelo RIVA (ÉCOLE D'ÉCONOMIE DE PARIS)

Table of Contents

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

Introduction

In this report, we present the results of two data connection case studies. The goal of the two case studies was to investigate the applicability of various techniques presented in Deliverable 6.2 (Cule, 2020b) on the task of connecting data present within the EURHISFIRM consortium to data coming from external sources. The first study, performed by the Antwerp team, mainly focused on using data matching techniques to establish links between the SCOB database and the London Share Price Database Monthly dataset (LSPM), which has been made available to us by its director Mike Staunton (London Business School). The second study, performed by the Paris team, focused on using a collaborative environment to register links identified between the DFIH database and the EUROFIDAI database in the Wikibase format. A detailed description of all databases involved in these case studies is included in Deliverable 4.4 (Poukens, 2019).

Case 1: Linking SCOB and LSPM

The goal of the study was to identify companies whose stocks were traded on both the Brussels and the London Stock Exchanges at the same time. More precisely, given that the available data is incomplete (the LSPM dataset is only a sample of the full data)[1], the goal was to find companies that were present in both datasets. To do this, we utilise some of the data matching techniques introduced in Deliverable 6.1 (Cule, 2020a) of this project, and evaluate their performance.

Experimental design

As established by our previous case study described in Milestone 6.1 (Cule et al., 2020), the matching of companies is best done using the normalised Levenshtein distance (Levenshtein, 1966) when comparing the names of the companies in the two datasets. The Levenshtein distance is a measure of how different two pieces of text (or strings) are. If the two strings are exactly the same, the Levenshtein distance will be equal to 0; if not, the Levenshtein distance is equal to the minimum number of steps (inserting a letter, deleting a letter or changing a letter) required to turn one string into the other. The normalised Levenshtein distance computes the Levenshtein distance relative to the length of the input strings. In this case, the distance can be normalised using either the shorter or the longer string.

Our goal is to identify as many as possible correct matches (or true positives) in the data, while avoiding incorrect matches (or false positives). We evaluate the performance of our algorithms by having the output verified by a domain expert. To minimise the human effort, it is therefore of paramount importance to avoid false positives. Naturally, the smaller the distance, the better the match. It is therefore logical to select only the candidate matches with the smallest distance for human inspection. We therefore examine

[1] For the period from 1955 to 1974, the LSPD includes a random sample of 33 percent of the companies listed on the London Stock Exchanges plus a sample of the largest quoted companies. From 1975, the database includes all listed companies (Staunton, 2019).

several distance thresholds to see how the algorithms perform. 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. 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.

Since the LSPM dataset was provided to us in a MS Access database, and the SCOB data is stored in an Oracle database, we first extracted the names of all companies present in the two databases for further analysis. As companies occasionally change their names, we also exported the start and end date for each company name. As a result, we obtained two csv files, containing the company ID, company name, start date and end date, for all companies present in the two databases. The SCOB database yielded 16733 entries, and the LSPM database 22431 entries. The matching algorithms were then implemented in Python and took the two csv-files as input. None of our matching algorithms were case-sensitive.

Evaluating the Thresholds

In this section, we evaluate the performance of our techniques at various distance thresholds.

Exact Matches on Corporation Name

As a first experiment, we identified corporations that have exactly the same name in the two databases. This produced 11 corporations, and 12 corporation names since we found two exact matches for one corporation (both 'Ladbroke Group PLC' and 'Ladbrokes PLC' were present in both databases). For illustration, we provide a subset of the output in the table below:

EXAMPLE	NAME	SCOB.ID	LSPM.ID
1	**Whitbread Plc**	9912	5595
2	**English Electric Company**	9974	1826
3	**Galapagos NV**	20039	11559

Our domain expert (Frans Buelens of the University of Antwerp) successfully verified that all 11 identified matches were correct. In other words, setting the distance threshold at 0 resulted in a true positive rate of 100%. However, it was also quite clear that we did not even scratch the surface in terms of finding all the possible matches in the data. It was therefore important to experiment at higher distance thresholds.

Varying the Threshold

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 systematically increase the distance threshold, while keeping an eye on the true positive rate. In our first experiment, we used Levenshtein distance normalised with respect to the longer string. Normally speaking, we would

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

expect the true positive rate to decrease as the threshold is lifted, until, at some point, the number of false positives becomes too large for human inspection (or the number of true positives too small to be worth the effort).

We first raised the threshold to 0.05, which resulted in just one additional match, namely ‘Standard Bank of South Africa’ in SCOB and ‘STANDARD BANK OF SOUTH AFRIC’ in LSPM. Clearly, this match was correct, and the distance was due to a typo. Raising the threshold further to 0.1, we obtained another three possible matches, one of which proved incorrect:

SCOB.ID	SCOB.NAME	LSPM.ID	LSPM.NAME	CORRECT
10176	Anglo Group PLC.	7755	ANGLO GROUP PLC	Yes
10922	NDS Group Plc	8908	IDS Group plc	No
11811	Cadbury Schweppes	985	Cadbury-Schweppes	Yes

Raising the threshold to 0.15 produced another 11 possible matches, of which seven were correct. At this threshold, the total true positive rate was thus 21/26, or 80.77%. An interesting observation was that we, for the first time, encountered two possible matches for one corporation: the ‘Baron Corporation’ in LSPM had a relatively small distance to both ‘Aon Corporation’ and ‘Enron Corporation’ in SCOB.

This final observation was a sign of trouble to come. Raising the threshold to 0.2 produced another 76 possible matches, of which only ten were correct. However, 31 of the incorrect matches concerned ‘NDS Group Plc’ in SCOB, which was considered similar to any three letter combination followed by ‘Group Plc’ in LSPM, as long as one of the three letters was ‘N’, ‘D’ or ‘S’. A further eight possible, but wrong, matches were found for ‘RHJ International’ in SCOB, for the same reason. Raising the threshold further to 0.25 resulted in another 472 possible matches, of which 263 involved ‘NDS Group Plc’. It became clear that going through such a list would not justify the required human effort. At the same time, even at this threshold, there were still many correct matches to be found. To do this in a manageable way required a different approach.

Searching for the Best Match

The problem described above stemmed from many, typically short, company names in one database being matched to a similar name in the other database. To avoid this problem, we decided to generate just one possible match per company name, namely the best match (or several best matches in case of ties), and ignore all others. Concretely, for ‘NDS Group Plc’ we would propose ‘IDS Group plc’ at a threshold of 0.1, and then never propose another match at any other threshold. The inherent risk in this strategy is that we might miss out on some true positives if the correct match is, for whatever reason, not the best match present in the data. On the other hand, by eliminating hundreds (and at higher threshold thousands) of

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

spurious matches from the output, we could dig deeper and discover matches that would have otherwise remained undiscovered.

It should be noted that this approach requires differentiating the roles the two databases play in the process. Producing the best match in LSPM for every company in SCOB and producing the best match in SCOB for every company in LSPM does not give the same results. Returning to the above problem, the best match for 'NDS Group Plc' in SCOB is 'IDS Group plc' in LSPM, but there will still be hundreds of companies in LSPM for which 'NDS Group Plc' in SCOB is the best match. In other words, the direction in which we perform the matching can greatly affect the results.

We performed four sets of experiments in this setting. We varied the direction of the matching process, and we also evaluated two ways to normalise the Levenshtein distance – normalising using the longer string and normalising using the shorter string. In the interest of fairness, in cases when two best matches were found, and one of them turned out to be correct, we count this as 0.5 true positive and 0.5 false positive.

A summary of the results (in terms of the true positive rate) is presented in the table below, with the first experiment included for comparison:

THRESHOLD	SCOB – LSPM (longer)	SCOB – LSPM (shorter)	LSPM – SCOB (longer)	LSPM – SCOB (shorter)	First experiment
0.05	12/12 = 100%	12/12 = 100%	12/12 = 100%	12/12 = 100%	12/12 = 100%
0.10	14/15 = 93.33%	14/15 = 93.33%	14/15 = 93.33%	14/15 = 93.33%	14/15 = 93.33%
0.15	21/26 = 80.77%	17/22 = 77.27%	21/25 = 84%	18/22 = 81.82%	21/26 = 80.77%
0.20	28.5/52 = 54.81%	23/39 = 58.97%	32/111 = 28.83%	26/95 = 27.37%	31/102 = 30.39%
0.25	39.5/115 = 34.35%	32.5/91 = 35.71%	N/A	N/A	N/A
0.30	41/211 = 19.43%	36.5/162 = 22.53%	N/A	N/A	N/A

The results show quite clearly both the importance of the direction of the matching and the value of the best-match approach. With the best-match approach we were able to dig deeper into the data by raising the distance threshold, thus discovering matches that would have been left undiscovered using other approaches, due to the unmanageable quantity of false positives in the output. In terms of the direction of the matching process, it seems sensible to generate the best match for every company in the smaller database, rather than the larger. This choice is not only supported by the results, but is also intuitive, as, regardless of the actual number of discovered matches, the larger database will, per definition, always have a larger number of remaining unmatched entries.

On the other hand, the choice of the normalisation string was not as significant. Naturally, for any match of two names of different length, normalising with respect to the shorter string results in a larger distance. As a result, the number of both proposed and true matches at any threshold is always larger (or equal)

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

when normalising using the longer string. However, the true positive rates at similar thresholds remained similar, from which we conclude that both normalisation methods are equally effective.

To conclude this section, we report a few more examples of correct matches that were discovered at high distance thresholds:

SCOB.ID	SCOB.NAME	LSPM.ID	LSPM.NAME
9909	Courtaulds Ltd	1408	Courtaulds plc
9514	Rio Tinto Plc	4345	RIO TINTO ZINC
11807	Hansen Transmissions	12407	Hansen Transmissions Intnl
10929	African Lakes Corporation	6514	AFRICAN LAKES CORP.

An Alternative Experiment

In all our experiments so far, both in this case study and the one described in Milestone 6.1, we considered datasets in which relatively few matches could be found. These were valuable experiments in the context of the data we possess at this stage of the EURHISFIRM project. However, as the project progresses, and the data become more and more integrated, and thus more and more complete, we will be facing an altogether different scenario. Given a complete integrated EURHISFIRM database, if a new dataset becomes available (either a new historical source or entirely new data), it is highly likely that our database will already contain some records that match the new data. In other words, unlike in the previous experiments, we would then expect to find a match in our database for (almost) any entity in the new data.

In our final experiment, we attempt to simulate this scenario. Concretely, we used corporation names from the director's names supplement to the 1915 edition of the Belgian 'Receuil financier' (a yearbook with information on the issuers of securities listed on the Brussels Stock Exchange) and compared them to the names in the SCOB database. This supplement lists the names of directors (*administrateurs*) and statutory auditors (*commissaires*) in alphabetical order with reference to the board positions they held in one or more corporations. Normally speaking, all those corporations should already be present in the SCOB database, but the names in 'Receuil financier' were often spelled differently or abbreviated (see example below). Standard abbreviations such as "Chdf." (chemins de fer) and "Hauts F." (hauts fourneaux) were resolved through automatic substitution before the experiment. This allowed us to test how our techniques would perform in a setting where expectations were much higher. In fact, if we make the assumption that all these companies were present in the SCOB database, we could for the very first time see which correct matches we actually missed out on. Such feedback is of paramount importance as we attempt to improve and fine-tune our methods.

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

Example of a name record in the director's names supplement of the *Recueil financier*:

Adriaensen, Louis, Anvers. — A. Chdf. Méridionaux d'Espagne. — Crédit National Industriel. — Ghezireh Estates. — Pétroles de Boryslaw. — Westende-Plage. — C. Hauts F. Aumetz-La Paix — Hauts F. de Fontoy.

The full 'Receuil financier' dataset contained 1252 company names. We started by using the Levenshtein distance, normalised with the longer string, with the threshold set to 0.1. This produced a match for 280 companies, all of them correct. Already this first experiment showed how different these results were to a setting where most of the data was expected to remain unmatched. However, successfully matching 280 out of 1252 companies was hardly satisfactory. This was mainly due to the 'Receuil financier' data sometimes being severely abbreviated. However, when we raised the threshold to 0.2, the results considerably deteriorated. Of the hundreds of proposed matches, the majority was incorrect, many of whom involved the same name being matched to multiple other names. We then attempted to use the best-match approach described above to discover exactly one match (with possible ties) for each company name in 'Receuil financier'. This, too, did not produce satisfactory results, as most best matches were clearly wrong. The reason for this was again the abbreviated nature of the 'Receuil financier' data in combination with normalisation using the longer string, which often resulted in company names being matched with short names that shared some generic terms, rather than the more unique aspects of company names. For example, 'Aciéries de Longwy' was matched with 'Aciéries de Mons', while the correct match would have been 'Société des Aciéries de Longwy'. In conclusion, the Levenshtein distance is a good tool to identify similar names at low distance thresholds, but struggles when abbreviations are used or entire words omitted.

The intuition behind our next approach can be directly illustrated by the example above. The name 'Aciéries de Longwy' is in fact entirely contained within the longer version 'Société des Aciéries de Longwy'. Naturally, the condition that one name must be entirely contained within the other is far too strict. It does not allow for abbreviations (other than of the final word) or for spelling errors. We therefore decided to look for the longest common subsequence (LCS) between the two names (Hirschberg, 1977). The longest common subsequence of two strings is defined as a sequence of characters that can be found within both strings, allowing for gaps but preserving the order.

The improvement in the results was dramatic. The best match based on LCS proved correct for over 80% of the company names. Furthermore, for some companies for which this approach did not result in the correct match, the actual match had already been discovered using the Levenshtein distance. However, there still remained a considerable number of unmatched companies, for a variety of reasons. In some cases, it was clear what went wrong. For example, some short names were, entirely accidentally, completely contained within some very long names in the SCOB database, resulting in a larger LCS score than with their actual match, which may have differed by one or two characters. In other cases, the usage of diacritical symbols caused a mismatch.

We therefore first cleaned the two datasets by removing all points, commas, etc., and converting all accentuated characters into their basic form (e.g., 'é' into 'e'). We then ran an approximate version of the LCS algorithm, whereby we produced the shortest match that had an LCS score of at least 90% of the

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

optimal score. Using this method, in combination with the previous efforts, saw us discover a match for over 90% of the data, leaving just 107 companies unmatched.

What stood out among the 107 unresolved cases is that they were often very short company names. For example, ‘Citas’ was matched to ‘Equitas’ using the Levenshtein distance, and to ‘Crédit Lyonnais’ using the LCS method. However, the correct match was ‘Compagnie industrielle et de transports au Stanley-Pool (Citas) (1907 - ...)’. Clearly, neither the Levenshtein distance nor LCS are suitable to find such well hidden matches. We therefore turned to an even simpler method – searching for cases where one name was a substring of the other (i.e., completely contained, with no gaps). This approach yielded another 22 correct matches, bringing the total to 1167, with 85 companies remaining unmatched.

A further inspection of the SCOB database revealed that for 37 of those 85 there was in fact no match to be found, which reduced the original sample to 1215. An analysis of the 48 unmatched companies revealed a variety of different reasons for the failure of the matching algorithms to find the correct match. In some cases, the abbreviations were too severe (e.g., ‘Automobiles SAVA’ and its correct match ‘Société Anversoise pour la Fabrication de voitures automobiles’), in others the order of the words was different (e.g., ‘Chemins de fer meridienaux italiens’ and ‘Société Italienne pour les chemins de fer méridionaux’), and some were simply too different to be matched by any algorithm (e.g., ‘Ciments North’ and ‘North's Portland Cement and Brick Works’).

These cases aside, we managed to correctly match 1167 out of 1215 companies, a total of 96.05%, which is highly satisfactory. Nevertheless, it is important to note that we needed a variety of sometimes very different techniques to identify all of these matches. The inevitable conclusion is that no technique is sufficient on its own. For nearly exact matches, Levenshtein distance performs well. For strings of considerably different lengths, LCS-based techniques give the best results, yet sometimes produce some glaring omissions, too. Finally, substring-based methods can help find very short strings in much longer strings and thus discover further matches.

In cases where we expect to find matches for (nearly) all data in one of the datasets, we conclude that an amalgam of approaches is needed. Ideally, an interactive interface should be designed that recommends to the user potential matches for each item (e.g., a company) in the dataset. The first recommendation could be based on the Levenshtein distance if it is low enough. The second might be the best match using the LCS-method (or its approximate version). The third might be generated by the substring method. The following recommendations could then be the second-best match from one of the methods, etc. Once the recommended match is accepted by the user (or if enough recommendations are rejected), no more recommendations would be generated for that item. This would allow the user to maximise the number of discovered matches, while minimising the effort. Concretely, in our experiment, the first recommendation would be correct in over 80% of the cases, and at most four recommendations would be needed to identify 96.05% of the correct matches.

Case 2: Linking DFIH and EUROFIDAI

Goals

This case study will demonstrate "data connecting" between two databases – DFIH and EUROFIDAI – using Wikibase. This is a follow-up to the case study described in Milestone 6.1 (Cule et al., 2020), in which we demonstrated "data matching" between DFIH and SCOB. In that previous report, we first introduced Wikibase as a collaborative environment to import, edit and use data from EURHISFIRM internal databases such as SCOB and DFIH. We then used Wikibase to demonstrate how to register and publish matches between companies from the two databases using a simple user interface.

As part of D6.2 Report on data connecting issues and methodologies (Cule, 2020b), we also defined the distinctions between "connecting" and "matching" as well as "data" and "metadata". Below, we reproduce these definitions, which are important to understand the technical and organisational implications for a future implementation of EURHISFIRM's work.

Data sources

DFIH, Data for Financial History, is a project at the Paris School of Economics, which created and maintains a database containing historical information on companies and stocks traded at the Paris Stock Exchange. Being part of the EURHISFIRM consortium, the DFIH database is defined as *internal*, in much the same way as SCOB.

EUROFIDAI, European Financial Data Institute, is a project funded by the French National Center for Scientific Research (CNRS), which created and maintained a database containing information about companies and stocks traded in 37 European countries, including France. EUROFIDAI is not part of EURHISFIRM; as such, it is a good example of *external* data. As part of this report, we have the authorisation to use and publish its metadata, which will be enough to demonstrate "data connection".

We also build upon our ongoing work on a dedicated instance of Wikibase for EURHISFIRM, which currently makes use of sample data from the DFIH and SCOB databases containing information about:

- Persons: IDs, name, gender, positions held and relevant dates.
- Companies: IDs, names, legal forms, locations, addresses and relevant dates;
- Stocks: IDs, name, emitting company and relevant dates.

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

Definitions

Internal vs External data: Data from consortium members such as DFIH or SCOB is considered "internal" to EURHISFIRM, while data from other sources (EUROFIDAI, London Stock Exchange) is said to be "external".

Connecting: We use the term "*connecting" (or its synonym "linking")* within the context of the EURHISFIRM project to define the process of establishing a conceptual link between at least two companies or stocks belonging to separate data repositories.**Matching**: While "connecting" is with data external to EURHISFIRM, "matching" is with internal data. As part of our work and for all intents and purposes, the differences are very limited in a technical sense. However, they carry a heavy weight in terms of governance and organisational decisions to make, which in turn will have consequences on the implementation infrastructure.

Data vs Metadata: Metadata can be understood simply as "data about data". Within EURHISFIRM's context, an example of metadata is the unique identifier and the name of a stock while the price of that stock (or rather the time series of its evolution) represents the actual data in that instance. The difference between data and metadata is an important one to make. Indeed, while the most comprehensive and successful matching methodologies can only be designed using complete exports from databases, connecting (or *linking*) can be achieved by using only metadata exported or even merely exposed from databases. Since EUROFIDAI is not open data, only its metadata can be used and published by EURHISFIRM. Consequently, it is a good example for data connecting.

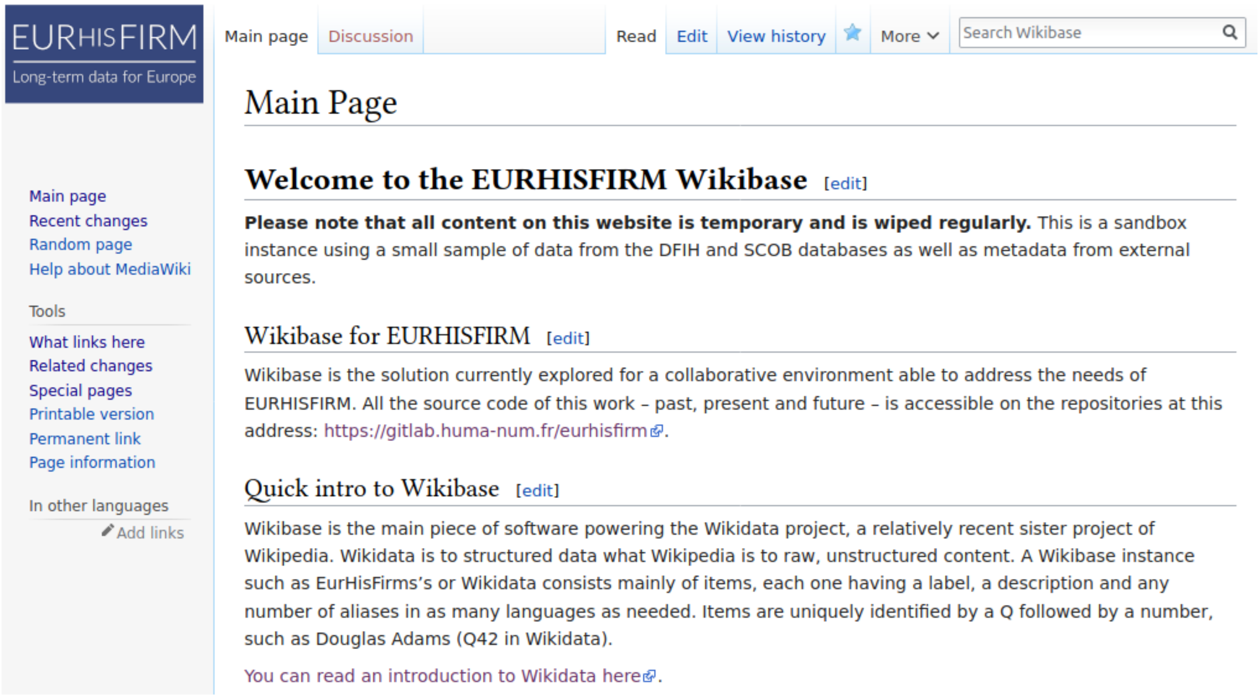
Wikibase

Wikibase is the solution currently explored for a collaborative environment able to address the needs identified by EURHISFIRM's members, namely:

- Merging, Matching & Connecting entities of all types (companies, stocks...) between all sources (internal as well as external);
- Enriching the data, i.e., editing the information via a user friendly interface;
- Visualising and exporting the results for the diverse purposes of the consortium members.

To be clear, Wikibase would not provide by itself the tool to *find* the matches and connections between entities in separate databases (internal or external) but would enable researchers to register and *share* them in a way that can be used by others. Consequently, many matching/connecting techniques and processes could be thought out, experimented and used independently by any parties. They would be run outside of Wikibase, the findings would be automatically registered by a program (bot) and then verified by humans, in a centralised and open database on the web, which would be the EURHISFIRM Wikibase platform.

The sandbox instance of Wikibase used for this work is accessible at this address: data.eurhisfirm.eu. Please note that this is currently used as an experimental and development test bed and is not intended for end users.



Main Page

Welcome to the EURHISFIRM Wikibase [edit]

Please note that all content on this website is temporary and is wiped regularly. This is a sandbox instance using a small sample of data from the DFIH and SCOB databases as well as metadata from external sources.

Wikibase for EURHISFIRM [edit]

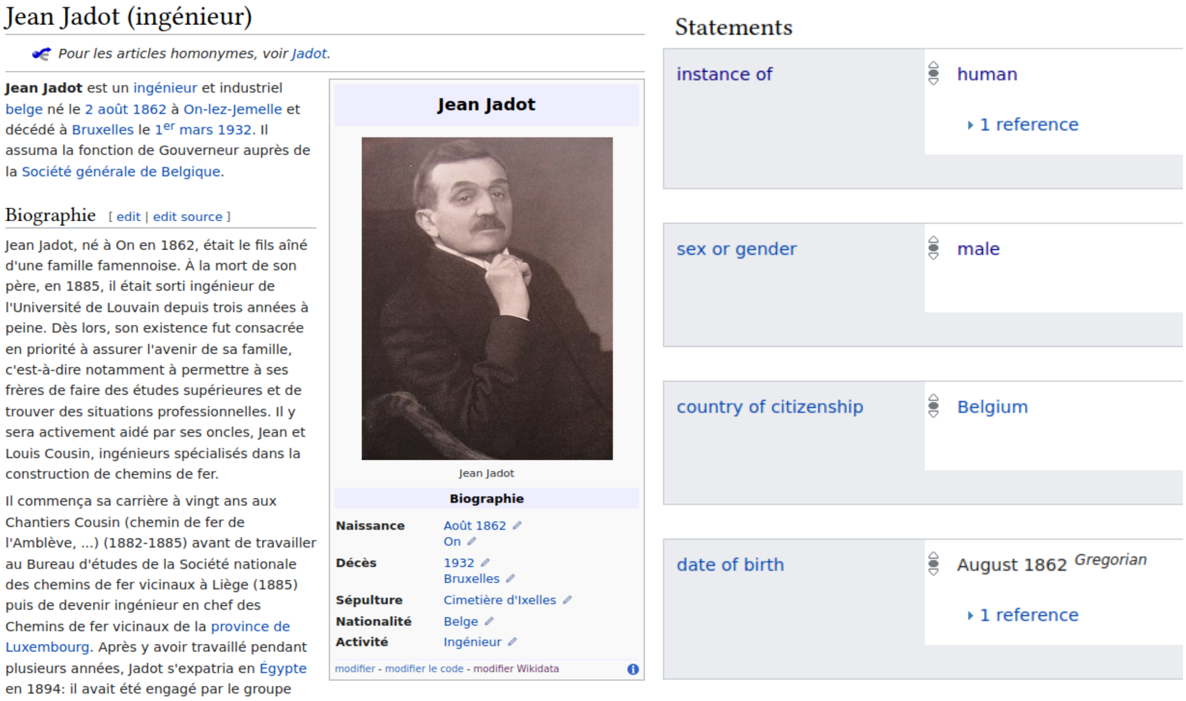
Wikibase is the solution currently explored for a collaborative environment able to address the needs of EURHISFIRM. All the source code of this work – past, present and future – is accessible on the repositories at this address: https://gitlab.huma-num.fr/eurhisfirm.

Quick intro to Wikibase [edit]

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. A Wikibase instance such as EurHisFirms's or Wikidata consists mainly of items, each one having a label, a description and any number of aliases in as many languages as needed. Items are uniquely identified by a Q followed by a number, such as Douglas Adams (Q42 in Wikidata).

You can read an introduction to Wikidata here.

Homepage of the EURHISFIRM Wikibase instance at data.eurhisfirm.eu.



Jean Jadot (ingénieur)

Pour les articles homonymes, voir Jadot.

Jean Jadot est un ingénieur et industriel belge né le 2 août 1862 à On-lez-Jemelle et décédé à Bruxelles le 1er mars 1932. Il assuma la fonction de Gouverneur auprès de la Société générale de Belgique.

Biographie [edit | edit source]

Jean Jadot, né à On en 1862, était le fils aîné d'une famille famennoise. À la mort de son père, en 1885, il était sorti ingénieur de l'Université de Louvain depuis trois années à peine. Dès lors, son existence fut consacrée en priorité à assurer l'avenir de sa famille, c'est-à-dire notamment à permettre à ses frères de faire des études supérieures et de trouver des situations professionnelles. Il y sera activement aidé par ses oncles, Jean et Louis Cousin, ingénieurs spécialisés dans la construction de chemins de fer.

Il commença sa carrière à vingt ans aux Chantiers Cousin (chemin de fer de l'Amblève, ...) (1882-1885) avant de travailler au Bureau d'études de la Société nationale des chemins de fer vicinaux à Liège (1885) puis de devenir ingénieur en chef des Chemins de fer vicinaux de la province de Luxembourg. Après y avoir travaillé pendant plusieurs années, Jadot s'expatria en Égypte en 1894: il avait été engagé par le groupe

Jean Jadot

Jean Jadot

Biographie

Naissance	Août 1862 On
Décès	1932 Bruxelles
Sépulture	Cimetière d'Ixelles
Nationalité	Belge
Activité	Ingénieur

Statements

instance of	human 1 reference
sex or gender	male
country of citizenship	Belgium
date of birth	August 1862 Gregorian 1 reference

Comparison of the interfaces of Wikipedia (left) and Wikidata/Wikibase (right).

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

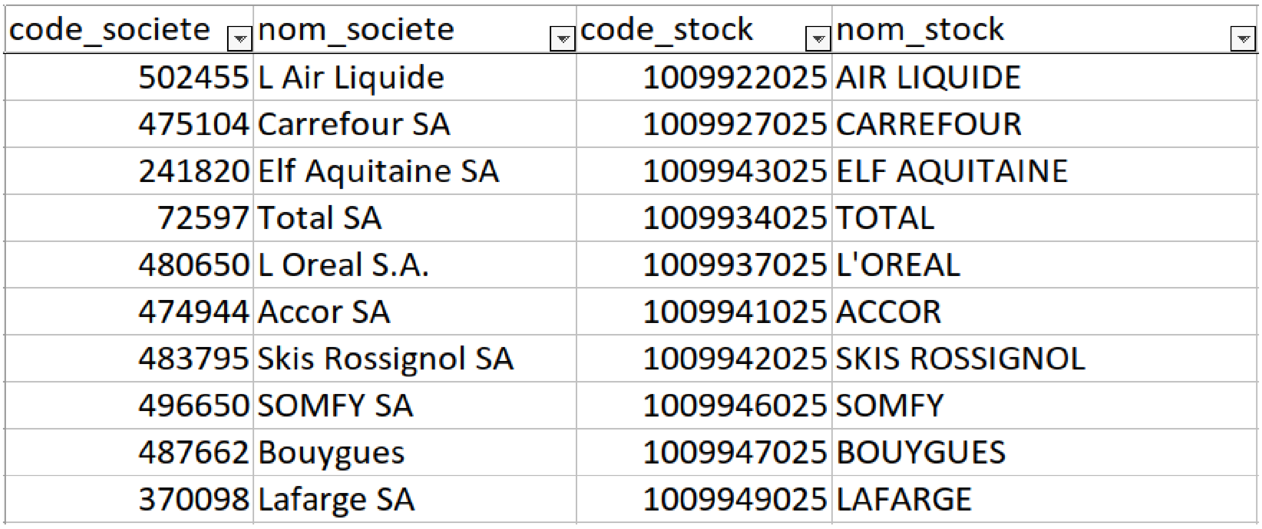
Steps

The case study consists of three steps:

1. Preparation;
2. Import;
3. Connection.

Preparing the data

The first step is to prepare the source data – EUROFIDAI, DFIH – before importing it into Wikibase. Once the data is exported from the source databases (see the "Data sources" section above), we have to verify and clean the samples that will be used for the case study. The exported data consists of tabular files in the CSV format.



code_societe	nom_societe	code_stock	nom_stock
502455	L Air Liquide	1009922025	AIR LIQUIDE
475104	Carrefour SA	1009927025	CARREFOUR
241820	Elf Aquitaine SA	1009943025	ELF AQUITAINE
72597	Total SA	1009934025	TOTAL
480650	L Oreal S.A.	1009937025	L'OREAL
474944	Accor SA	1009941025	ACCOR
483795	Skis Rossignol SA	1009942025	SKIS ROSSIGNOL
496650	SOMFY SA	1009946025	SOMFY
487662	Bouygues	1009947025	BOUYGUES
370098	Lafarge SA	1009949025	LAFARGE

Sample data extracted from the EUROFIDAI database, here presented in tabular form.

Data from the EUROFIDAI project used for this case study contains stocks traded on the Paris Stock Exchange. Particularly useful for us are:

- Name and EUROFIDAI ID of the emitting company;
- Name and EUROFIDAI ID of its stock.

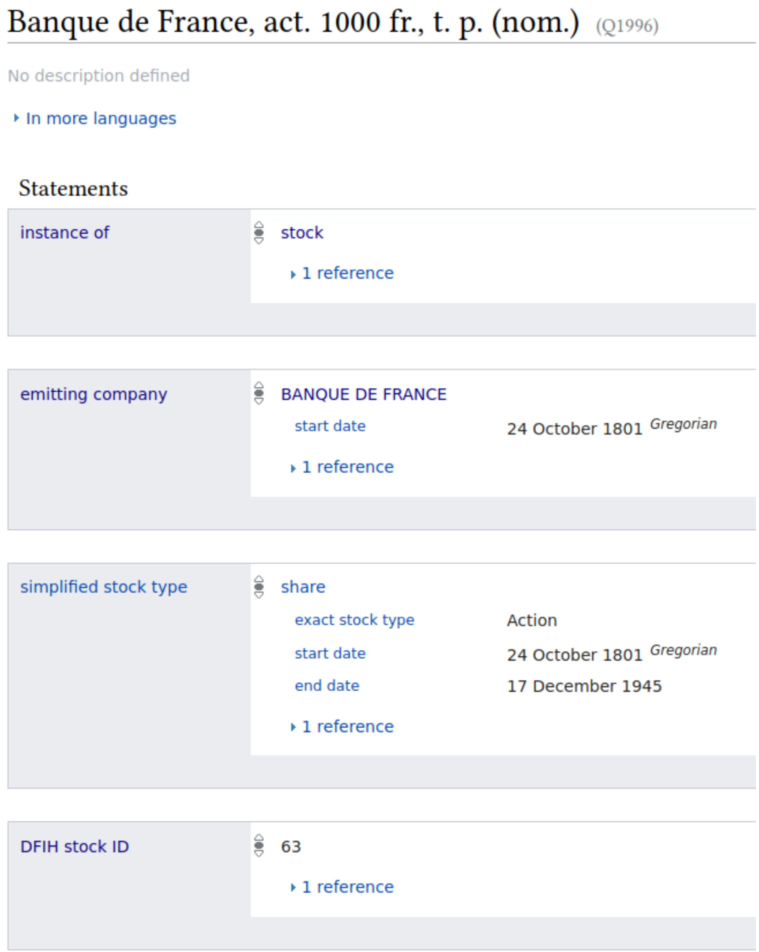
Data from the DFIH database contains similar information on stocks traded on the Paris Stock Exchange:

- Names of the stock, as extracted from different yearbooks;
- Emitting company identified with a DFIH Corporation ID;
- DFIH Stock ID;
- Stock type.

Importing the data

Once the files containing the samples from EUROFIDAI and DFIH are cleaned and ready, we can import the data to Wikibase. There are many different ways to add data in Wikibase, manually or automatically. While tools with user interfaces such as OpenRefine, QuickStatements or Mix'n'match exist, they do not provide the flexibility we needed for our work. For this case study, 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 to the samples from the DFIH and EUROFIDAI databases in order to correctly represent the source data from its original format (tables, columns, line...) to the Wikibase model (items, properties, values...).

The result of the import is the creation of a Wikibase page ("item") for every single stock and company sampled from the EUROFIDAI database, as well as for stocks and companies from DFIH. On Wikibase, the connection between a stock and its emitting company, which we can see for example in the sample from EUROFIDAI capture above, is modelled using a property we called "emitting company". The nature of each item (stock or company) is then identified thanks to the dedicated "instance of" property. This data modelling applied during the import phase is intended to structure the data in a way that allows future complex usages based on SPARQL queries.



Banque de France, act. 1000 fr., t. p. (nom.) (Q1996)

No description defined

▸ In more languages

Statements

instance of	stock	
	▸ 1 reference	
emitting company	BANQUE DE FRANCE	
	start date	24 October 1801 Gregorian
	▸ 1 reference	
simplified stock type	share	
	exact stock type	Action
	start date	24 October 1801 Gregorian
	end date	17 December 1945
	▸ 1 reference	
DFIH stock ID	63	
	▸ 1 reference	

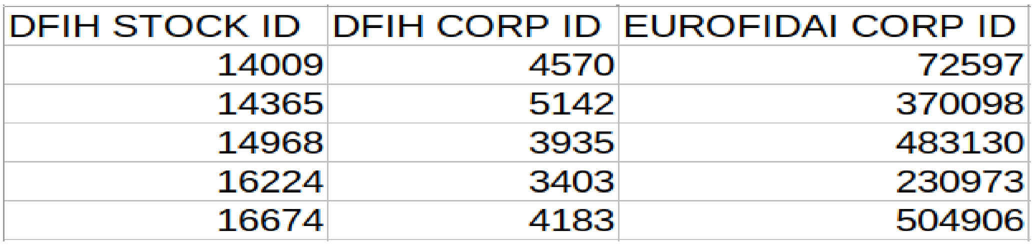
Example of a stock item imported in Wikibase from the DFIH database.

Connecting the data

Finding connections between items – companies, stocks and persons – from separate databases can be done using many different methodologies, some of which have been explored in the first case study described in this report. For our part, we will not delve into complex techniques and only use the few existing cases of companies and stocks with exactly the same name which are easy to find. This allows us to rather focus on the added value of Wikibase itself and articulate how those findings would be actually integrated and published in the platform.

With Wikibase, registering connections between two items, for example a company from DFIH and the same company in EUROFIDAI, is as simple as *merging* those items. A merge effectively combines all the information from the two pages into a single one without losing the traceability and with the ability to always revert those changes.

In the previous case study described in Milestone 6.1 (Cule et al., 2020), we showed how a user can manually merge two matched items using the Wikibase built-in tool with a simple user interface. Now, we will present a solution to automate this process in order to merge multiple items at once.



DFIH STOCK ID	DFIH CORP ID	EUROFIDAI CORP ID
14009	4570	72597
14365	5142	370098
14968	3935	483130
16224	3403	230973
16674	4183	504906

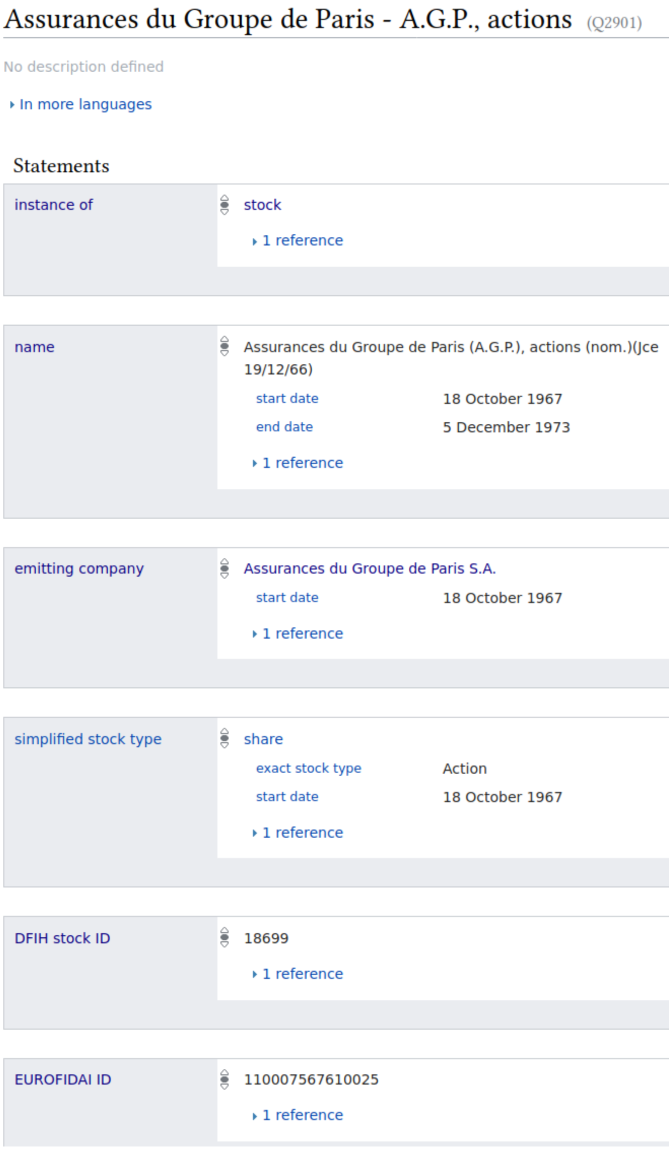
Example of a simple file containing connections between an internal database (DFIH) and an external database (EUROFIDAI).

Thanks to its extensive tooling and the flexibility of its API, Wikibase allows us to build a program (a “bot”) able to take a simple CSV file with connections as input (see example above) and register them by directly merging the corresponding items. For this use case, we consider that the connections have been verified and confirmed as exact before giving it to the bot. A more advanced workflow implementation could add a verification step, with a dedicated interface built upon Wikibase, in order for other researchers to confirm the findings before the merging step.

The bot takes the IDs from the source databases and finds the items with those IDs already imported in Wikibase (see previous step). It also uses the connections between companies and stocks already modelled. It then merges items with IDs that were connected. Once the bot as finishes the merges, we can verify that the data connecting from DFIH and EUROFIDAI is correct.

Results

In the following screen capture, we see a single page in Wikibase (or “item” as it is called in this case) that describes a stock identified with a EUROFIDAI Stock ID as well as a DFIH Stock ID. Note that this item is itself identified by an identifier, Q29001, which is unique to this Wikibase instance. All items merged to this one still keep their own Q identifier but are now automatically redirected to this one. Note also that the title at the top the page, called the item "label", can be edited as well as translated in different languages. This is just a human-readable label on top of the machine-readable identifier, and is displayed to the user on Wikibase depending on the language they configured. This is not used to actually store any structured metadata; for this we use a dedicated property (here simply "name") which can have multiple values (indeed a corporation often changes names) as well as qualifiers (date, location...) and references (source database, yearbook, etc.).



Assurances du Groupe de Paris - A.G.P., actions (Q2901)

No description defined

› In more languages

Statements

instance of
stock
› 1 reference

name
Assurances du Groupe de Paris (A.G.P.), actions (nom.)(Jce 19/12/66)
start date 18 October 1967
end date 5 December 1973
› 1 reference

emitting company
Assurances du Groupe de Paris S.A.
start date 18 October 1967
› 1 reference

simplified stock type
share
exact stock type Action
start date 18 October 1967
› 1 reference

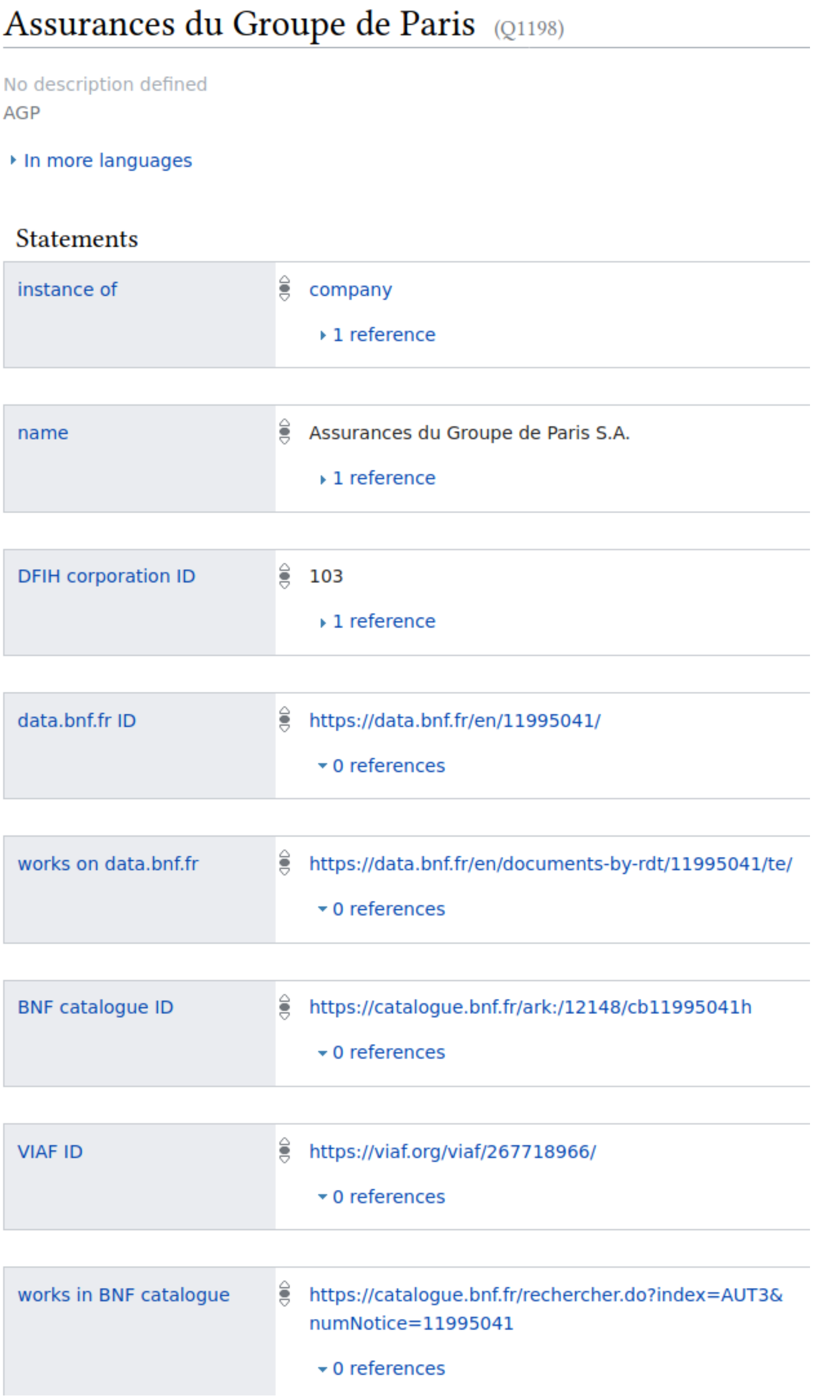
DFIH stock ID
18699
› 1 reference

EUROFIDAI ID
110007567610025
› 1 reference

The two databases – DFIH and EUROFIDAI – are now "connected".

We can connect as many databases as we want: indeed, the more external IDs are present, the better the data connection. For stocks, SICOVAM, RGA and CCDVT IDs also come from DFIH, while more recent ISIN IDs are given by EUROFIDAI. EURHISFIRM Legal Entity Identifiers (ELEI) and Financial Instrument Identifiers (EFII) would be added in the same way.

As a final demonstration, we can also improve the data connection in the form of “web linking”, meaning that we can link, with a URL, a company to an external authority file published on the web. Here, for example, is the company corresponding to the stock presented above:



Assurances du Groupe de Paris (Q1198)

No description defined
AGP

‣ In more languages

Statements

instance of	company ‣ 1 reference
name	Assurances du Groupe de Paris S.A. ‣ 1 reference
DFIH corporation ID	103 ‣ 1 reference
data.bnf.fr ID	https://data.bnf.fr/en/11995041/ ▾ 0 references
works on data.bnf.fr	https://data.bnf.fr/en/documents-by-rdt/11995041/te/ ▾ 0 references
BNF catalogue ID	https://catalogue.bnf.fr/ark:/12148/cb11995041h ▾ 0 references
VIAF ID	https://viaf.org/viaf/267718966/ ▾ 0 references
works in BNF catalogue	https://catalogue.bnf.fr/rechercher.do?index=AUT3&numNotice=11995041 ▾ 0 references

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

Conclusion

In this report, we described two case studies that deal with the challenges of connecting data present within the EURHISFIRM consortium to data available from external sources.

The first case study, performed by the Antwerp team, attempted to identify links between the SCOB database and the LSPM dataset, containing data from the London Stock Exchange. The case study investigated a number of data matching techniques in terms of the true positive rate, as well as the number of spurious potential matches sent to the domain expert for verification. We further studied different strategies in linking the two datasets. Of particular importance was the analysis of the importance of the direction of the best-match search techniques, leading us to the conclusion that approaching the problem from the “wrong” direction can lead to serious reduction in performance.

While the SCOB and LSPM data produced very few matches, we expect the future EURHISFIRM database to be well populated, such that newly arriving data will typically already have a match in the historic database. To simulate this scenario, we performed an additional experiment, which showed that no single technique is capable of identifying all matches in the data. We conclude that an ideal method would provide users with recommendations originating from a variety of underlying data matching algorithms, such that the correct match can quickly be found.

The second case study, performed by the Paris team, examined how connections between various databases can be integrated within the Wikibase format. These concepts were tested and evaluated by connecting the DFIH and EUROFIDAI datasets. The case study resulted in the development of an intuitive and user-friendly interface allowing users to easily inspect the data and register further findings. Furthermore, automatic methods were developed for merging items in Wikibase based on the results of other data matching algorithms.

References

Cule B. (2020a). EURHISFIRM D6.1: Report on data matching issues and methodologies. University of Antwerp, Antwerp

Cule B., Buelens F., Poukens J., Annaert J., Richer J. (2020). EURHISFIRM M6.1: Data Matching Case Study. University of Antwerp, Antwerp

Cule B. (2020b). EURHISFIRM D6.2: Report on data connecting issues and methodologies. University of Antwerp, Antwerp

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

Hirschberg, D. S. (1977). Algorithms for the longest common subsequence problem. Journal of the ACM (JACM), 24(4), 664-675

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)

Poukens, J. (2019). EURHISFIRM D4.4: Report on data and sources documentation and quality assessment. University of Antwerp, Antwerp

Staunton, M. (2019). London Share Price lspm201812 & lspd201812 Reference Manual. London Business School, London

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