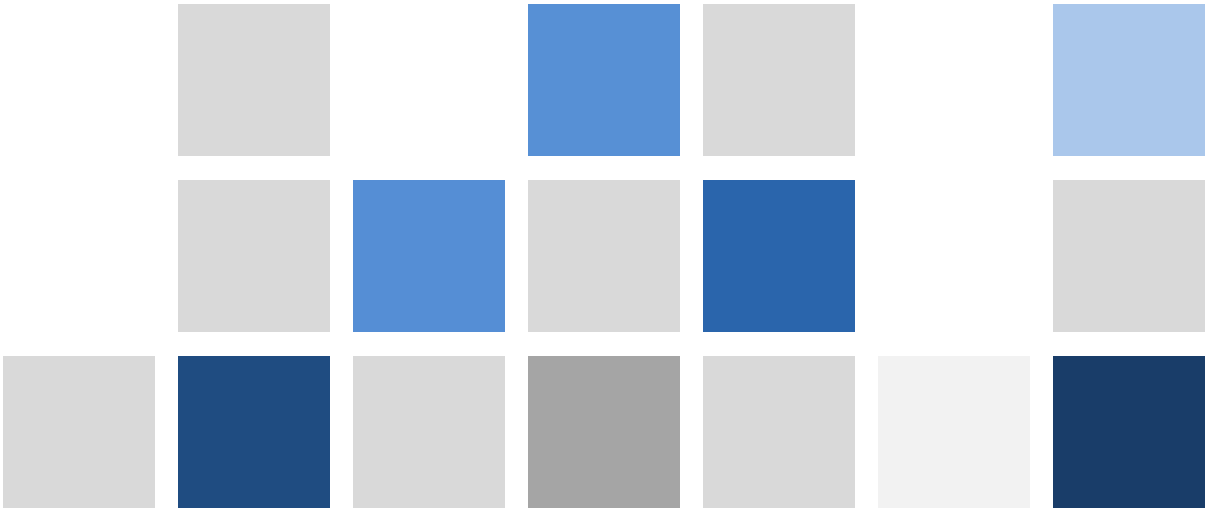
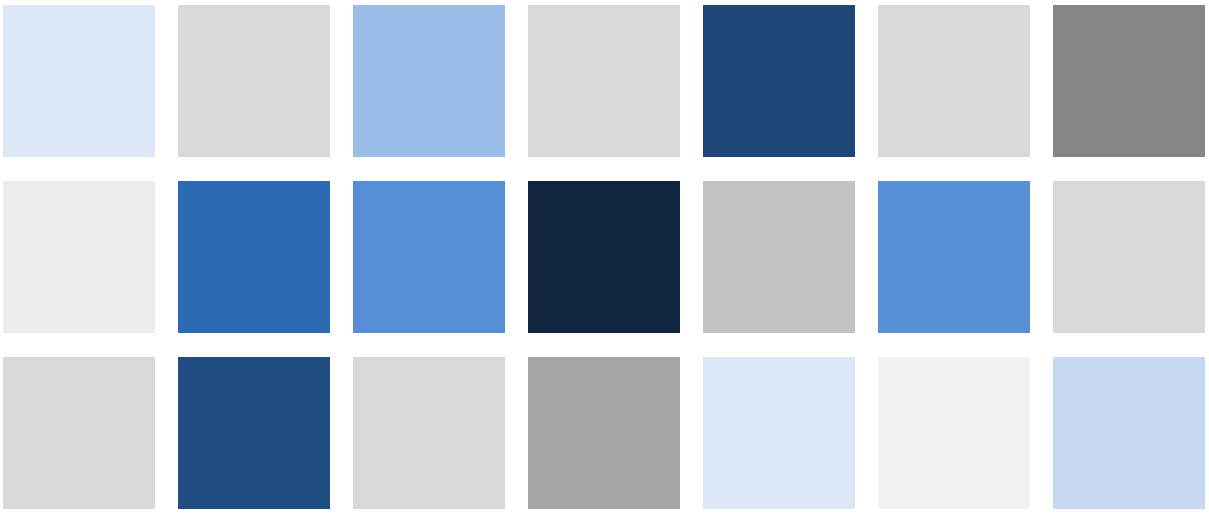


Long-term data for Europe

EURHISFIRM

D6.1: Report on data matching issues and methodologies



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

AUTHOR(S):

Boris CULE (University of Antwerp)*

APPROVED IN 2020 BY:

Jan ANNAERT (University of Antwerp)

Wolfgang KÖNIG (Goethe University)

Angelo RIVA (Paris School of Economics)

* The author would like to thank Angelo Riva (Paris School of Economics), Johan Richer (Jailbreak, Paris), Coen Fierst van Wijnandsbergen (Rotterdam School of Management, Erasmus University), Johan Poukens (University of Antwerp) and Lana Yoo (Paris School of Economics) for their valuable feedback that led to the successful completion of this report.

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

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 document, we discuss the concept of data matching, as well as the issues that arise from attempting to match data from various sources within the EURHISFIRM project. Furthermore, we present an overview of existing data matching methodologies and discuss the extent to which they are, either directly or after suitable adaptations, applicable within our framework. We also discuss a possible methodology to integrate databases and the outputs of the matching procedures within a collaborative environment, allowing us to enrich and harmonise datasets in the perspective of cooperating at the European level.

We begin this report by describing the available data, identifying the crucial differences in data formatting and storage among various sources within the consortium that make the data matching task particularly challenging. We then present a selection of the state-of-the-art data pre-processing techniques that should be used where appropriate in order to facilitate the later process of data matching.

In the central part of the report, we discuss a variety of data matching techniques specifically designed for particular data matching problems. We start with the issue of schema matching, where, given two (or more) databases, the goal is to identify which tables in different databases may contain the same (or similar) data; and, in a second step, which attributes (or columns) in these tables should be matched to each other (and crucially, how). Once the schemas have been matched, we can proceed with record matching, where the goal is to identify which records (or rows) in one table can be matched to other records in another table (in the other database).

The main goal of data matching is to correctly identify the data that should be matched; but in the presence of large quantities of data, this should ideally be done efficiently, using automated techniques whenever possible. However, efficiency often comes at the cost of accuracy, and we investigate to what extent we can rely on automated techniques while ensuring the correctness of the output. Additionally, further methodologies allow iterative matching over time between multiple databases, while also allowing those databases to keep evolving by themselves. We argue that in some cases, human intervention is necessary to verify (or reject) the possible matches discovered by data matching algorithms. This perspective leads to the rationale for creating a collaborative environment where human efforts and automatic techniques can register matches between entities in separate databases to improve the outputs of the tools that find those matches.

Finally, we discuss the benefits of the results of a successful data matching process for both data analysis and data cleaning purposes. We conclude the report with a summary of the identified issues involved in data matching and the respective solutions applicable to the EURHISFIRM project and its specific needs.

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

Background

Within the EURHISFIRM consortium, we have access to a variety of datasets. These datasets all contain information about companies traded on the stock exchange on a national level. The individual datasets, as it stands, are not only physically stored in different locations but also come in a variety of storage formats.

The most mature datasets are those of the Studiecentrum voor Onderneming en Beurs in Antwerp and of the Data for Financial History in Paris, or SCOB and DFIH, respectively. These two datasets are both stored in similarly designed Oracle databases using a relational data model. Even though the original design for the two databases was essentially the same, there is some divergence between the current database schemas of the two datasets.

Other datasets available to the consortium are not as advanced as SCOB and DFIH. Data from Germany, Spain and the UK mostly consist of spreadsheets, each structured and formatted in a different way. In this context, we examine the issues that arise when attempting to match data coming from these different sources, as a crucial step towards a unified European data repository.

A more detailed overview of the available datasets is provided in Deliverables 4.3 and 5.1.

Data Pre-processing

Before we can begin with the data matching task, some data pre-processing may be necessary.

A major pre-processing issue is that of data formatting, since the available datasets come from a variety of sources in a variety of different formats. For example, the SCOB and DFIH datasets are stored in Oracle databases, while other data is provided in raw Excel files. However, most data matching techniques rely on the data being stored in the same format. In practice, the selected format could be a relational database (such as Oracle), simple text files (such as CSV) or some other structured format (such as RDF). For example, since our two most populated and most reliable datasets already are in the Oracle database format, we could attempt to convert all other datasets into this format as well. In some cases, such a conversion would be straightforward (e.g. if the Excel file is properly formatted, with each separate sheet containing data that would, after conversion, make up a single database table, and with column headings that could be used as attribute names in Oracle). In other cases, the Excel files are less structured, and specifically designed software may be needed to complete the conversion.

While data formatting ensures that all the data is stored in the same format, this is not enough to secure a smooth data matching process. A further step that is required is data harmonisation. Data harmonisation tackles issues such as variability in data type or range, even in data coming from comparable sources. Concretely, a data matching algorithm may struggle to match numerical data with an attribute stored as text, even if that textual attribute in fact contains only numbers. It is therefore important to ensure that all data tables use appropriate data types, and that data types are used consistently in all tables. For example, even if the two attributes that need to be matched are both numeric, but one uses an integer as data type and the other allows decimal numbers, this would be undesirable.

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

Another pre-processing technique that could be of interest is data synchronisation. The data we work with is temporal in nature, and we need to make sure that the reported time stamps coming from various sources are synchronised and reliable. For example, if datasets come from different time zones (such as Paris and London), and they contain hourly data, we should be using a universal clock for any matching task rather than relying on local time. Equally, to facilitate data matching, there should be conventions on what the recorded time stamps actually mean – for example, is the "end date" of a corporation the date on which the corporation ceased to operate, or the day after (i.e. the first day on which the corporation was no longer operational)? Alternatively, the databases could be exported to a new semantic format. Because of their flexibility, semantic databases do not need strict data pre-processing work, allowing the source-databases to be kept alive in their original formats and ensuring regular synchronisation among the databases.

Data Matching

In this section, we discuss the main topic of this report, the issue of data matching. The data matching process consists of two steps – schema matching and record matching – which we describe below.

Schema Matching

Given two or more databases, the goal of the schema matching procedures is to identify which tables in various databases contain similar information and then to identify which columns in those tables can be matched. In other words, the goal is to determine which columns are semantically related. For example, the CORPORATION_NAME table in the SCOB database and the CORPORATION_NAME table in the DFIH database contain similar information, but not all columns can be matched. Concretely, the corporation IDs in the two tables do not match at all, but an attempt can be made to match corporation names.

While there are automated techniques for schema matching (Rahm et al, 2001), this task within the EURHISFIRM project is fairly straightforward and can be performed manually, given the readily available expert knowledge about the respective datasets. Automated techniques would require large pre-processing efforts and time. Any discovered matches would per definition be uncertain and would still require verification by a human expert. Within the framework of semantic databases, this process would be done through the matching of entities and properties related to a common ontology.

Record Matching

Once we have identified which columns in which tables from various databases can be matched, the next step is to attempt to match actual data records in those tables. For example, while the CORPORATION_NAME tables in the SCOB and DFIH databases may contain similar information, not all data records in those tables can be matched. Concretely, many corporations that are listed in the SCOB database do not appear in the DFIH database and vice versa. Our goal is to identify those that appear in both.

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

In some cases, the record matching task can be relatively simple and will only require looking for an exact match in the respective columns. In other cases, multiple attributes may need to be matched simultaneously. For example, to find a match between two people in two different databases, it is typically not enough to find an exact match on the NAME attribute, but this should be also searched on the DATE OF BIRTH attribute. Sometimes, further checks may be necessary.

However, in more complex cases, looking for exact matches is nowhere near sufficient. For example, if a corporation is listed in both the SCOB and the DFIH databases, it may well be that its name is written differently in the two respective tables. In such cases, we need techniques that identify uncertain, or partial, matches. Such techniques rely on distance metrics between individual attribute values or even between combinations of attribute values. Examples of distance measures are edit distance between strings (such as the Levenshtein distance (Levenshtein, 1966), or the Jaro-Winkler distance (Winkler, 1990), or simple differences between numeric or date/time attributes, or specially designed distance metrics defined on specific data types (for example, dynamic time warping can help measure the distance between two time series (Berndt and Clifford, 1994)). Other metrics may need some pre-processing of the raw attribute values, such as removing stop words (or other unnecessary terms) from strings, or using abbreviations instead of full strings. The very same algorithms under very different programming languages could use the semantic databases that can also import the output of matching performed within relational databases though SQL queries.

Efficiency

While the schema matching task can, in our case, largely be performed manually, the record matching task relies on automated procedures. As discussed earlier, for such procedures to work, it is practical that the tables to be matched are stored in the same databases, at least temporarily. The data matching queries must, in theory, compare every entry in one table with every entry in the other table, looking for potential matches. This, of course, is only the worst-case complexity of such algorithms, which can be significantly reduced depending on the context. For example, when looking for exact matches, the data could be sorted alphabetically and then scanned in parallel, leading to a much faster matching process. If the data is structured hierarchically, record matching can be performed level-wise, identifying matches at a higher level before proceeding to the lower levels. For example, if we first match corporations in two databases, then we can used this matching information when attempting to match individual stocks. Concretely, instead of comparing all stocks present in the two databases to each other, we only need to compare the stocks of the corporations for which we have found a match in the other database. However, even when optimised, these can be very time-consuming operations that should ideally not be performed on data in different databases, and especially on data in different physical locations that rely on an internet connection, for example.

In some cases, the issue will be exacerbated by the fact that multiple tables in different databases may be needed for the matching task. For example, if the same stock of the same corporation is traded at various stock exchanges, it may be necessary to not only compare the names of the corporations and of the individual stocks in the two databases, but also to compare the actual historic prices of the stocks at the two stock exchanges. If it really is the same stock, one would expect the stock price of the stock to be similar (though probably not exactly the same) at different stock exchanges at the same time. To ascertain

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

such a match, we would need data from the tables containing corporation information, general stock information, temporal stock price information, and even temporal exchange rate information. Such a massive join would be very time consuming even in the most efficient database engines and would be entirely unmanageable if the tables were stored in separate databases.

However, once the data has been matched, the data in the original databases that has been used in the matching process should remain unamended. In other words, the matching process should have read-only access to the data it relies on. The results of the matching process can be used to build indices that contain the matching information. A fictional example of such an index is shown below:

SCOB.CORPORATION.ID	DFIH.CORPORATION.ID	STARTDATE	ENDDATE
25	49	1/1/1900	31/12/1999
66	78	5/12/1947	17/11/2011
…	…	…	…

This index would tell us that the corporation stored under corporation ID 25 in the SCOB database is in fact the same corporation that is stored in the DFIH database under corporation ID 49. Note that the results of the data matching process are also temporal in nature. Any discovered match is only valid within a defined period in time. In the example above, corporation 25 in the SCOB database may be matched with another corporation in the DFIH database prior to 1/1/1900 or may not have a match at all.

In the final step of the matching process, such an index can be stored in each of the corresponding databases, allowing them to enrich their own respective data with information coming from other sources, but it could also be stored entirely externally. For example, these indices could facilitate an external interface relying on its own query engine which could be used to simultaneously query all databases. In this case, it would be useful to introduce an overarching EURHISFIRM ID to uniquely identify an entity (for example, a corporation). A global index would then match this unique ID to various local IDs in national databases. Relying on this global index, a unified interface could then send subqueries to the individual databases based on the data matching information contained in the indexes, before accumulating the results and producing an integrated output.

The same kind of indexation could be performed within semantic databases that can also import the indices produced by the matching between two relational databases, if already done. As a consequence, the global index could be provided, for example, within an instance of Wikibase. An instance of Wikibase is a kind of database with human and machine interfaces to enable data recording and collaborative contributions via the web. In this way, as many matching processes as needed can be created, even by third parties, run in parallel outside of Wikibase and their findings automatically registered by a program (bot) in Wikibase. The global index would be then updated according to the new matching process and human contributions.

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

Evaluation

In the most abstract terms, the data matching task can be described as a case of binary classification (Tan et al., 2016). For each pair of records, we need to determine whether they can be matched or not. Our goal is to classify each pair of records correctly. When relying on automatic procedures for binary classification, there are a number of ways the performance of such procedures can be evaluated. The simplest evaluation measure is accuracy, defined as the percentage of instances that have been classified correctly. In our case, however, accuracy is nearly meaningless, since most pairs of records clearly should not be matched. In these circumstances, an algorithm optimised for the accuracy metric would simply assign a "no match" label to each pair of records and easily achieve very high accuracy.

This problem is typical for cases of imbalanced classification, where one class is dominant. In such cases, other evaluation measures can be more informative. To begin with, we can divide the output into four groups: true positives (the pairs matched by an algorithm that should have been matched), true negatives (the pairs not matched by an algorithm that should not have been matched), false positives (the pairs matched by the algorithm that should not have been matched), and false negatives (the pairs not matched by the algorithm that should have been matched).

In principle, the goal is clearly to maximise the true positives and negatives and to minimise the false positives and negatives. However, in cases of imbalanced data, the true negatives are innumerable and therefore not very informative. This is why alternative evaluation measures focus on the remaining three groups. *Precision*, defined as the proportion of true positives within the group of instances that have been classified as positive ($p = TP / (TP + FP)$), measures how well the method is performing in terms of avoiding false positives; while *recall*, defined as the proportion of true positives within the group of instances that should have been classified as positive ($r = TP / (TP + FN)$), measures how well the method is performing in terms of avoiding false negatives (Powers, 2011).

Typically, aiming to simultaneously maximise both *precision* and *recall* is contradictory. Recall can be boosted by classifying more instances as positive, but this also usually increases the number of false positives, decreasing the precision. Similarly, to improve precision, the number of false positives should be reduced, but this usually results in more false negatives, decreasing the recall. The F1-score, defined as the harmonic mean of precision and recall, can be used to optimise on both measures simultaneously, but this is only desirable when giving equal importance to both precision and recall (Hand and Christen, 2018). In our case, however, it is clear that false positives are a lot more harmful than false negatives. If a wrongly identified match finds its way into the system, it can immediately lead to erroneous query results and further data analysis. A false negative, on the other hand, does no direct harm, and could potentially be identified as a match later on. We conclude that we should err on the side of caution, and focus on eliminating the false positives.

Human Supervision

While fully automated record matching techniques have been a topic of theoretical research for a long time (Newcombe et al., 1959), in practice, such approaches have been successfully used only in very limited settings (He et al., 2018; Abramitzky et al., 2019), relying exclusively on exact matches or on known

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

and fixed data structures. In general terms, an automated method will evaluate each potential match using a certain similarity measure and then simply output the matches that have a similarity score higher than some predefined threshold. Clearly, in our setting, where false positives can be very harmful, this approach would be too risky.

Nevertheless, even with this in mind, record matching algorithms remains crucial to our efforts, as they can be used to rank the potential matches based on a similarity score and offer them to a human expert for verification. In this way, the human effort is minimised, as the algorithm can relatively quickly discard the majority of pairs of records that clearly cannot be matched. However, we cannot rely on an algorithm to automatically insert matches into the newly-built indices without expert verification.

Aside from the verification of the discovered matches, human supervision may prove useful in other steps along the way. To begin with, expert knowledge is key to a successful pre-processing of the data. Furthermore, depending on the data that is to be matched, domain experts can define new or fine tune existing distance metrics between data records, as well as decide which metrics to use in specific individual cases.

Above, we have discussed how human supervision is necessary to ensure that only correct matches find their way into the integrated database. While automated data matching procedures are needed to efficiently process huge amounts of data, we must also keep in mind that some matches may well be identified without using such algorithms. Clearly, a human expert may possess enough information about a particular company to insert a matched pair of corporation IDs into the database, even if the respective individual datasets do not currently possess enough information for an algorithm to discover the match. With this in mind, we should provide additional interfaces for humans to manually register known matches into the database. Moreover, we do not need to limit ourselves to the data present in our datasets, since additional data may well be available from other publicly accessible sources online. Such data may be valuable in helping us identify further matches, which could then be integrated into the EURHISFIRM database by either humans or algorithms (bots). As mentioned above, these tools could easily be implemented within a Wikibase environment.

Iterative Process

In a dynamic setting, the data matching task never ends. The datasets used within the EURHISFIRM project are not static, and new data is constantly streaming in. This new data can be current data (such as daily stock prices), but also historic data inserted into the databases as new sources are discovered or processed. Moreover, errors could be found in the current versions of the database and corrections could be needed. For each new record in any single national database, new matches could potentially be found in the remaining datasets. Additionally, a change in an existing record could also result in a new match or could even invalidate an already discovered match.

There are two ways to approach the issue of dynamic databases. One is to perform live checking and trigger a data matching algorithm every time a record has been inserted or amended in any single dataset. This, however, would be very costly, and the possible benefits would not justify the effort. The alternative is to periodically rerun the matching algorithms to check if any new matches can be discovered or if any

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

matches stored in a previous iteration have been invalidated. Such periodic reruns could either be scheduled at regular temporal intervals or be triggered by a fixed predefined amount of changes in the data. However, for crucial updates in the databases, such as a change in a corporation ID, instantly triggering an update of the matching information would be required.

An additional benefit of regularly repeating the data matching process is that, due to new data, matches that may have been missed in previous iterations may well be discovered by later iterations, thus reducing the overall number of false negatives. Alternatively, the repeated processes could be automatically run by bots within the framework of semantic databases embedded in Wikibase environments.

Data Post-processing

The available EURHISFIRM datasets are mostly sparsely populated. The data that is already there does not contain any contradictions or internal conflicts, but there is no guarantee that the data is entirely correct and clean. The data matching task, while being important in and of itself, can also be helpful in identifying potential errors in the data and in even in suggesting potential corrections where possible. Therefore, we end this report with a short discussion of data cleaning procedures that could be applied after the data matching task.

Data cleaning methods can broadly be divided into two categories: interactive and automatic. Interactive data cleaning methods identify possible instances of erroneous data and present them to the user, who makes a final decision on whether the data is accurate or not and, if necessary, takes action to remove or correct the encountered errors. Automatic data cleaning methods attempt not only to identify erroneous data but also to automatically correct the errors. While interactive data cleaning methods are safer (as no data will be removed or amended without direct input from the user), they require much more human effort and time. However, automatic methods are typically considered too risky for usage on raw data. These methods include, for example, identifying outliers in a time series and replacing them with a value using interpolation of neighbouring values. In stock exchange data, such outliers may well contain correct data on days when some major event influenced large fluctuations in stock prices. Clearly, it would be undesirable if any algorithm made such “corrections” to the EURHISFIRM data. However, the interactive data cleaning methods such as outlier detection are of interest even without the data matching task, but these remain outside of the scope of this report. It is nevertheless important to note that these kind of corrections would easily be implementable within the Wikibase environment through additional bots that could also propose corrections on given parameters.

Within our context, once the data matching process is completed, it can unearth potential errors in the data. For example, if the same stock of the same company is traded at various stock exchanges in various currencies, it would still be expected that the price at which the stock is traded (after currency conversions) would be similar everywhere. Therefore, if one of the datasets contains an erroneous price for a particular stock on a given day, this error would be very difficult to spot in a single dataset. However, relying on the results of the data matching task, we could use the fact that the two stocks have been matched to check

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

whether the two prices also match. If not, the interactive data cleaning tool would ask the user to inspect the two records and determine which one should be corrected.

Summary

The major data matching issues and possible solutions are summarised in the table below.

	Issue	Solution
1.	Different data formats in different datasets	Convert all datasets to the same format • e.g. an Oracle database • e.g. a semantic database (such as Wikibase)
2.	Different data types used for storing similar data	Data harmonisation: use consistent typing/ ontologies
3.	Data coming from different time zones	Data synchronisation: use universal clock
4.	Different conventions on the meaning of temporal data	Fixed conventions to be used across all datasets
5.	Schema matching	Done manually by domain experts
6.	Record matching	Using automated techniques with human supervision and verification within a collaborative environment
7.	Similarity/distance measures	Specific to data types
8.	Efficiency	Reduce time complexity through simplifying problems
9.	Accuracy	Focus on precision, rather than recall
10.	False positives	Should be avoided
11.	False negatives	Discover in future iterations
12.	New/updated data/correction of errors	Periodical reruns
13.	Final result	EURHISFIRM wide IDs linked to national databases through a global index

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 describe the main issues encountered when attempting to match data from various datasets within the EURHISFIRM consortium.

A major challenge lies in the fact that the original data comes in a variety of data models, formats and languages. However, the task of schema matching is still relatively straightforward, given the available expert knowledge about the structure of the respective datasets.

For the record matching task, on the other hand, automated techniques and collaborative environments are necessary. Given the quantity of the available data, it is important that the methods are efficient in terms of time and other resources, but not at the expense of accuracy. At the same time, since there is an inherent element of uncertainty involved in data matching, we cannot rely exclusively on algorithms. While an algorithm can be used to identify potential matches in the data, human supervision is necessary to verify which of those potential matches are in fact true matches and which are not. To facilitate and limit human efforts, it is therefore crucial that the algorithms rank the potential matches such that the most likely matches are evaluated by the human expert first.

The data matching task is a never-ending and iterative process. As new data comes in, more matches may be found and need to be identified. However, periodic reruns of the algorithm should suffice in most cases. The repeated processes could be automatically run by bots under human supervision within the framework of semantic databases embedded in the Wikibase environments.

In task 6.2 of the EURHISFIRM project, we will perform a data matching case study to evaluate the solutions proposed in this report on a subset of the available datasets. Furthermore, in Deliverable 6.2, we will discuss the task of matching the EURHISFIRM datasets with data coming from external sources such as EUROFIDAI. In that task, this report will be a valuable starting point, as many of the issues identified here will again be present with other more specific challenges that this new task will present us.

References

Abramitzky, R., Mill, R., & Pérez, S. (2019). Linking individuals across historical sources: A fully automated approach. Historical Methods: A Journal of Quantitative and Interdisciplinary History, 1-18

Berndt, D. J., & Clifford, J. (1994, July). Using dynamic time warping to find patterns in time series. In KDD workshop (Vol. 10, No. 16, pp. 359-370)

Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2006). Duplicate record detection: A survey. IEEE Transactions on knowledge and data engineering, 19(1), 1-16

Hand, D., & Christen, P. (2018). A note on using the F-measure for evaluating record linkage algorithms. Statistics and Computing, 28(3), 539-547

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

He, Z. L., Tong, T. W., Zhang, Y., & He, W. (2018). A database linking Chinese patents to China's census firms. Scientific data, 5, 180042

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)

Newcombe, H. B., Kennedy, J. M., Axford, S. J., & James, A. P. (1959). Automatic linkage of vital records. Science, 130(3381), 954-959

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation

Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. the VLDB Journal, 10(4), 334-350

Tan, P. N., Steinbach, M., & Kumar, V. (2016). Introduction to data mining. Pearson Education India

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

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