Long-term data for Europe

EURHISFIRM

D6.2: Report on data connecting 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 Johan Richer (Jailbreak, Paris) and Jan Annaert, Frans Buelens and Johan Poukens (University of Antwerp) 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 issue of connecting existing external data sources to individual databases belonging to the EURHISFIRM partners and, ultimately, to the central EURHISFIRM database, containing either the integrated data itself or links to other repositories where data is stored.

We study a wide variety of external data sources, coming in different formats, ranging from structured databases, Excel files or text files all the way to publicly accessible web pages. We identify the challenges of processing the data from such a variety of sources, before presenting state-of-the-art techniques for overcoming these challenges.

Before the EURHISFIRM databases can be linked to external sources, the external data needs to be gathered and analysed. The relevant records that can be linked must be identified, before the entities contained in both datasets can be matched to each other and a link established. For this task, methods introduced in Deliverable 6.1 can be reused, and further, more robust, methods will be developed.

Finally, once the connections have been established, we discuss how the linking data will be stored, and how the links will be maintained.

This report discusses a wide range of challenges, potential problems, and respective possible solutions, in a language that is approachable to a non-technical audience. At a later stage (Milestone M6.2), we will produce a report on the findings of our data connecting case study, in which we will present the technical details of the concrete technological solutions we applied and tested.

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. One goal of the EURHISFIRM project is to develop a common data model for a unified integrated EURHISFIRM database, which is tackled in Work Package 5.

On top of the datasets already in the consortium's possession, there is an enormous amount of data available from external sources. This can be historical data from stock exchanges not covered by the EURHISFIRM datasets, such as the Stockholm Stock Exchange, or contemporary data that has not yet been processed by the EURHISFIRM partners, such as the EUROFIDAI dataset or the London Share Price Database (LSPD). Such sources of data can prove very useful in both enriching and verifying the data we already possess.

The process of consuming and using data, be it from the consortium's national databases or from external sources, remains fundamentally the same. All data that EURHISFIRM will ever consume ultimately lives in a series of heterogeneous source databases that are administered and updated by independent third parties, either from inside or outside the consortium. Only the future implementation choices made by the consortium with regards to the EURHISFIRM infrastructure will determine how that data flow process will practically look like.

Since the goal of this report is to inform those strategic and technical decisions with a particular focus on the connecting aspect of the infrastructure, we need to consider what exactly do we consider as *connecting* and what is required to make the necessary connections. We define *connecting* within the context of the EURHISFIRM project as the process of establishing a conceptual link between at least two entities belonging to separate data repositories.

The difference between data and metadata is an important one to make at this point of EURHISFIRM's progress. Indeed, while the most comprehensive and successful matching methodologies can only be designed using complete exports from databases, linking (or connecting) can be achieved using only metadata exported or even merely exposed from databases.

Metadata can be understood simply as "data about data". Within the 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.

One of the main goals of EURHISFIRM can be summarised as creating a central point from which all available data can be accessed, either by directly consulting the data stored centrally or by following the established and stored links to data stored elsewhere.

However, before we are able to harvest the knowledge originating from external sources, we first need to analyse the data structure for each individual source, and subsequently identify which entities present in an external source can be linked to a counterpart already present in our data.

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

In this report, we discuss the four main phases of this process – gathering the external data, formatting it for further processing, matching the relevant entities when possible, and, finally, storing the results for future use.

Gathering the Data

External data can come either in a static or a dynamic form. Typically, historical data is static, and can be gathered in a single, one-off, procedure. Recent, new, and in particular future, data comes dynamically, and needs to be gathered constantly or periodically.

Much of the historical data is readily available in digital form, either provided to us directly by external partners, or through files located on dedicated websites containing historical information. Such data can come in the form of structured Excel files, csv-files or relational databases, in the wikibase format or in simple text documents. Naturally, the more structured the data, the less effort will be required for processing it.

Other historical data is only available on paper, in which case the documents must first be scanned, and then processed. In other cases, scanned versions of historical documents already exist and are available. Extracting data from such sources is the topic of Work Package 7 of this project.

However, whichever form the historical data may come in, its format remains fixed, and the effort of gathering it is limited to analysing the source, developing a software package for extracting the data where necessary, and then storing the data for further processing.

By far the most challenging scenario of gathering data from digital sources is that of collecting the data from websites, where the data is not provided within structured downloadable files, but on the web pages themselves. This process is called web scraping (Johnson and Gupta, 2012).

Web Scraping

Web scraping is the process of extracting data from web pages. Typically, the data to be collected is not contained on one single page. Before the data can be extracted, we therefore need to ascertain which web pages need to be accessed and analysed. The process of sequentially visiting a series of relevant web pages is called web crawling (Novak, 2004). A program that crawls the web automatically visiting a series of web pages is called a crawler. Crawlers can range in scope, from universal crawlers, that aim to visit and index the entire world wide web (such as those used by search engines), to targeted crawlers, that are specifically designed to access various web pages on a single website. The latter case is relevant within our project, if we identify a website that provides data that can enrich the data we already possess.

Unlike universal crawlers, that only serve to index the web pages present on the web, we need to combine crawlers with scrapers that will also extract the relevant data from each visited web page. In the best case, the data will be structured on the web page itself, and contained within a well-defined table.

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

This would enormously facilitate the extraction of the data, but it is not always the case. At the most basic level, the web page, downloaded in the HTML format, may need to be carefully parsed to determine the structure within it. A parser, however, must be robust, as web pages are rarely fully consistent, even within the same website.

Once a web page has been downloaded and parsed, only the relevant data needs to be extracted from it. A web page will typically contain a lot more than just the data that needs to be gathered. For example, there can be a title at the top of the page, links to other pages on the side, or contact details at the bottom, none of which may be of interest. To automate this process, the structure of the web pages needs to be carefully analysed, and specific software developed for each individual data source.

Additional Challenges

Both crawling and scraping are very challenging tasks. An important problem facing crawlers is avoiding visiting the same web page more than once (Liu, 2007). If a crawler visits a page and then continues by simply visiting all pages linked to from the first page, the next page will probably contain a link back to the previous page. This should be avoided not only to avoid the double work, but also because it would lead to infinite loops, ensuring the work is never finished. However, while it is not too difficult to avoid visiting the same page twice, this becomes more of a problem for scrapers if the same data is contained on more than one web page.

This problem is exacerbated in case of new data. For example, new data may be published on a website each day on the *same* web page. The crawler will then need to visit the same page daily, and the scraper will need to extract the data from the same page, which will be different from the data collected on the same page the previous day. To complicate things further, there may well be an overlap, and redundant data would need to be discarded.

Naturally, new data sources may appear in the future, and new software will need to be developed to extract the useful data from them. However, we must also keep in mind that an existing source may well change the way it provides its data to the public. This may occur through changes in website design, internal web page naming conventions, or additional types of data being made available. In such cases, the already developed crawlers and scrapers may no longer function properly, and it is important that they themselves somehow “realise” that, and alert the user to the need to redevelop them.

Finally, consulting external sources of data may also bring legal challenges. Websites may be protected by copyright and other intellectual property rights. Gathering the data for analysis is typically allowed in most cases. Storing the data, or storing links to the data should normally also not represent a problem. However, making the data originating from proprietary sources publicly accessible may raise legal issues. Such legal challenges were investigated within Work Package 3 and reported on in Deliverable D3.1.

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

Pre-processing the Data

Once the data has been gathered, it needs to be prepared for further use. Each data source comes in its own format, but for the task of data linking, the data needs to be as standardised as possible, ideally conforming to the common data model developed in Work Package 5 of this project.

The data pre-processing steps that are relevant here are much the same as those discussed in Deliverable 6.1 (Cule, 2020), that reported on data matching issues when dealing with data available within the EURHISFIRM project. In particular, we elaborated on three data pre-processing procedures.

First, data should be uniformly formatted. The chosen format must be suitable for the data linking task, and may well be separate from the format chosen for data storage in the later stage. For example, in our data matching case study (Milestone 6.1), we matched data using csv-files and the wikibase format, while the underlying data was stored in relational databases.

Second, the data should be harmonised. For various data items to be linked, they need to be of the same data type. For example, if the price of a security is for some reason stored in a text field, this could cause problems when trying to match it to a numerical attribute in another dataset.

Third, the data should be synchronised. Different data sources may store the date/time information differently, and data may be coming from a variety of time zones. Additionally, some date/time information may be more limited than others (for example, only the date may be provided, or, particularly in historical sources, even only the month and the year).

Linking the Data

Once we have collected the data from an external source, and appropriately formatted it, we can start with the main task – establishing the links between the data we already possess and the newly acquired data.

For this task, we partially rely on the data matching techniques discussed in Deliverable 6.1 (Cule, 2020). At the basic level, we need to discover which data items from one dataset correspond to certain data items in another dataset. Regardless of the number of available data sources, the matching of individual sources can be performed pair-wise, keeping in mind the transitive property of the identified matches. If item *a* from database **D1** has been matched with item *b* from database **D2**, which has then been matched with item *c* from database **D3**, then we know that items *a* and *c* also form a match. In the long run, once a central EURHISFIRM database is fully operational, it will suffice to link each new data source only to the central database, and not to underlying national databases.

The data matching process can be separated into two steps: schema matching and record matching. The goal of the schema matching procedures is to identify which tables in various databases contain similar

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

information, and then to identify which columns in those tables can be matched. While there exist automated techniques for schema matching (Rahm et al, 2001), this task within our project is straightforward and can be performed manually, given the readily available expert knowledge about the available datasets. Furthermore, any external sources will have necessarily been carefully analysed before the data has been gathered.

The goal of record matching is to ascertain which actual data items in various databases represent the same real-world entities. This task is much more complex and requires automated procedures. The task is made more onerous by the fact that in most cases, especially when datasets originate in different countries and, for example, contain data from different stock exchanges, the majority of records in each dataset will not match any record in other datasets. In these circumstances, we have to rely on algorithms to find the set of needles in a haystack – namely, the relatively few matches that may well be there (a typical example is cross-listing of shares, where a company lists its equity shares on multiple stock exchanges).

Naturally, no algorithm can find all correct matches without finding some incorrect ones. This is why human supervision remains a key ingredient of the data linking procedure. Any potential link between two data sources must be verified before it is taken to be certain. In the next two subsections, we first discuss how to identify possible matches in the data, and then how to present them to a human expert for verification.

Distance-based Techniques

In our earlier report on data matching techniques, we presented a number of techniques that can identify potential matches between data items in two different datasets. In a nutshell, these techniques attempt to quantify the difference between the two items. For example, if the two items are both numerical, we can simply compute the Euclidean distance between them, and if the two items are both pieces of text, we could compute the Levenshtein distance (Levenshtein, 1966) or the Jaro-Winkler distance (Winkler, 1990) between them.

However, in many cases, such simple distance measures are inadequate, and more specific methods may be required. For example, when attempting to match two people from two data sources, we may need to compare the first name and last name stored in the two sources as text fields, and, at the same time, take into account the date of birth, stored as date fields. To do this, we would need to define a unified measure that somehow incorporates individual distance measures used on individual data fields (for example, some sort of weighted average). Additional attention must be paid to handling missing data (for example, if the date of birth is missing, do we put the whole weight on the name, or do we introduce some sort of penalty?).

In some cases, especially if the data comes from different countries, further pre-processing steps may be required. For example, if matching company names, generic terms such as “Company”, “Corporation” or “Incorporated” might have to be removed, to avoid the computed distances being artificially inflated by comparing the same words in different languages. When comparing prices or dividends, on the other

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

hand, it would be necessary to convert them all into the same currency before any sensible data matching could take place.

In other cases, instead of comparing full strings contained within text fields, it may be better to compare them on a word-per-word level, keeping in mind that the order of the words may be different in different sources or languages. Alternatively, it may be of interest to identify the size of the overlap between the two strings, regardless of the differences in the non-overlapping parts. Techniques such as *longest common subsequence* (Hirschberg, 1977) and *longest common substring* (Arnold and Ohlebusch, 2011) can prove valuable in such cases, as they focus on what the two strings have in common, which can often be the defining element of a company name. In this context, the longest common substring approach identifies the longest *contiguous* sequence of characters in the two strings, while the longest common subsequence approach allows for non-matching characters to appear between the matched characters. In this sense, the longest common substring approach is much stricter, as it does not allow for spelling variations or typos, but, in the other extreme, the longest common subsequence approach can match sequences of characters that have nothing to do with each other, as they can be very far apart. Both of these methods can also be applied on a word-per-word basis.

Clearly, the choice of method will depend largely on the case in hand. The method chosen for matching company names will naturally be very different from the method chosen for matching security prices. Moreover, multiple methods may need to be used concurrently for the same task, as there will always be cases of outliers that even the best method cannot discover, that may yet be discovered by another method that does not perform as well for the more usual cases.

The performance of data matching methods can be evaluated using four indicators: true positives (the data item 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 such as ours, where the true negatives hugely outnumber all other categories, alternative evaluation measures that focus on the remaining three indicators are far more informative. 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). Finally, since it is impossible to simultaneously maximise both precision and recall, the F1-score is defined as the harmonic mean of precision and recall, and can be used to optimise the overall performance, but this is only desirable when giving equal importance to both precision and recall (Hand and Christen, 2018). In our case, false positives are a lot more harmful than false negatives. If a wrongly identified match finds its way into the system, it can lead to erroneous query results. A false negative, on the other hand, does no direct harm, and could still be identified as a match later on. We therefore give priority to methods that produce fewer false positives.

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

However, regardless of method and case, the final word will always be with the human expert, as no link can be considered reliable without expert verification.

Interactive Interface

The record matching algorithms, such as those discussed above and others, identify potential matches in the data. Each algorithm can be used to produce a ranking of possible matches that can then be offered to a human expert for verification.

In this context, it would be helpful to design an interactive interface which could be used by different human experts to evaluate potential matches and mark which are correct and which are not. Such an interface would have multiple advantages.

First of all, the interface could offer the expert possible matches produced by various techniques in parallel, and could quickly, through expert feedback, be able to evaluate the performance of each technique on the fly, and then gradually give priority to possible matches produced by the better-performing techniques.

Secondly, different users may specialise in different aspects of the data. One expert may want to verify potential matches between people, while another may wish to focus on securities. The interface should then provide the results of different matching techniques to each user.

Thirdly, an expert may wish to find a single specific match in two specific databases. In the general case of matching two databases, we may know in advance that for most entities no match will be found, but if we know in advance that in a particular case a match should (or even *must*) be found, that would significantly alter the approach. Instead of ranking all possible matches between the two databases, we should now, for a particular item (person, company, security) from one database, rank all possible matches in the other database. In this case, instead of only using a ranking produced by the best-performing method, we may now try to alternately provide suggestions produced by a variety of techniques (for example, if the first three suggestions produced by the best-performing method all turn out to be false, it may be better to try another method, as we might be dealing with a special case).

In summary, while the automated methods cannot be relied on to definitively ascertain which data items from different databases should be linked, they play a crucial role in minimising the required human effort. Ideally, the human expert will only need to verify that the proposed matches are correct. In this context, every false positive is a waste of the human effort needed to reject it. This, again, confirms the necessity of focusing on methods that produce few false positives.

Finally, the interface should allow the human to bypass the automated matching techniques as well. Clearly, a domain expert that possesses information about matching items from two databases should be able to simply insert such information into the integrated database even if such a match is not offered by any algorithm (in fact, even without consulting any algorithm in the first place).

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

Storing the Linked Data

Once we have linked our data to data coming from external sources, we need to decide what to do with the newly acquired information. In this section, we discuss two options – integrating the external data into our own database, or storing just the necessary linking information that would allow us to consult the external source for the extra information in the future.

Integrating the Data

In some cases, it may be perfectly possible to fully integrate the newly acquired data into the EURHISFIRM database. These are cases when the data obtained from an external source perfectly fits into the existing data model, requiring no amendments, other than inserting the new information. For example, if we have matched two corporations, but the external source contains information about the address of the corporation's headquarters that we do not yet possess, this address can easily be inserted into our database.

However, a full integration of the data may result in conflicts. For example, we may have matched two people by ascertaining that they were in fact the same person in two different databases, but the date of birth may be different in the two databases. In such cases, we may decide to import the external date of birth into our database *in addition* to the one we already have. In this case, it would be necessary to note the necessary information about the source of the data, so any researchers using the data can take this into account.

In other cases, fully integrating the external information with our database may not be possible without amending our data model. The external source may well contain the type of information our database was not designed to contain. In such cases, we would need to consider whether the necessary changes are even desirable.

Linking the Data

There can be a variety of reasons why we may choose not to integrate the external data into our own database. First of all, this might not be technically possible due to differences in the nature of the data. Second, if the source is stable and reliable, this might not be necessary. Third, if the external source is large, this may be too costly. And fourth, if the external data is proprietary, it may not even be allowed.

However, whichever case we are dealing with, what we can certainly store in our own database is the linking information. Namely, for any type of entity in our database (company, security, person, etc.), we can build an index containing the entity's ID in our database, a link to the external source, and the same entity's ID within that source. This information should suffice for us to be able to consult all the information about this entity from that source, as long as the source remains available and stable (e.g., if the IDs change, the links would need to be reestablished).

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 link data from external sources to the data already present within the EURHISFIRM consortium.

We consider a variety of possible types of sources, ranging from properly structured data files provided by external partners to data collected from publicly accessible websites. For the latter, we discuss technological solutions required to facilitate automatic collection of data from web pages, as well as the need for robust solutions capable of handling, or at least recognising, issues arising from possible changes in design of source web pages.

We further discuss how the data needs to be pre-processed before we can proceed with the main task of identifying links between data items present both in our data and in the external sources. The connecting process relies heavily on automated data matching algorithms, that can identify entities present in multiple databases that could potentially be linked. However, even the best algorithms are not fool-proof and a level of human supervision is required to ensure no false information is inserted into the database. Furthermore, an interactive interface should be developed to allow human users to register linking information independently of any automated procedures, too.

Finally, once the connections have been established, we present two alternative strategies for storing the newly acquired information. One option is to fully import the additional data into the integrated EURHISFIRM database, and the other is to only store the linking information necessary to access the data from the external source on the fly. We explore the advantages and disadvantages of the two approaches and evaluate which approach may be more appropriate in different scenarios.

The data connecting task is not a finite process. As both our data and most of the external sources will typically regularly be updated with new data, matches that were originally not there may appear later, and will need to be identified. Furthermore, new sources may become available, too. Periodic reruns of the algorithms should therefore be performed.

In task 6.4 of the EURHISFIRM project, we will perform a data connecting case study to evaluate the solutions proposed in this report on a subset of the available datasets. The resulting report will present concrete instantiations of some of the abstract technological approaches discussed in this report.

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

References

Arnold, M., & Ohlebusch, E. (2011). Linear time algorithms for generalizations of the longest common substring problem. *Algorithmica*, *60*(4), 806-818

Cule, B. (2020). D6.1: Report on data matching issues and methodologies

Johnson, F., & Gupta, S. K. (2012). Web content mining techniques: a survey. International Journal of Computer Applications, 47(11)

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

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)

Liu, B. (2007). *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media

Novak, B. (2004). A survey of focused web crawling algorithms. *Proceedings of SIKDD*, *5558*, 55-58

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

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