Data Extraction and Integration for the Creation of a Database Describing Portuguese Unions and Other Social Partners

João Manuel Ramires Machado

Abstract

The development of a complete and consistent database raises many challenges, related to data collection and integration from multiple sources. The DGERT institute is responsible for publishing a weekly official bulletin with detailed information on work-related organizations and on their activities. This institute also has information on trade-unions and employer organizations in different data sources.

In the context of this work, was developed a database on Portuguese labor unions and employer organizations, together with a web-based interface supporting the exploration of the data. The developed database contains information obtained from the different data sources available from DGERT institute. The database was developed through information extraction and data integration procedures, envisioning the support to different studies in the social sciences. For the development of the web-based interface, the Flask web framework was used.

The database development involved multiple small tasks. In the future it is possible to enhance those tasks, in order to improve the amount of data and the quality of the database.

1 Introduction

Empirical studies in the social sciences like Vandaele (2000) and Jensen (2020) frequently include or desire to include objective measures of unionization, including private and public sector labor union membership, density estimates, coverage, and representativeness. However, the development of a complete and consistent database focused on labor union membership raises many challenges, related to data collection and integration from multiple sources. In Portugal, the Boletim do Trabalho e Emprego (BTE) is a weekly official bulletin that publishes detailed information on work-related organizations like trade unions and employer organizations and on their activities. However, information extraction from the textual contents of the BTE is technically challenging, requiring the development of tailored approaches. Moreover, in support their day-to-day operations, the Direção-Geral do Emprego e das Relações de Trabalho (DGERT) institute that publishes the BTE also has information on trade-unions and employer organizations on relational databases and excel spreadsheets. Still, extracting the relevant information from these databases, and integrating it with additional contents extracted from the BTE, again requires the development of custom approaches.

The main goals of this work were to create a database with all the information regarding portuguese unions and social partners and develop a web application where that database would be used and manipulated. In order to create a database with all the information obtained from the data sources, the data sources were analysed and data cleaning and integration techniques were performed. Regarding the BTEs, they were analysed and a part of the important information gathered was integrated in the database. In order to DGERT employees manipulate and use the database, a web application was developed using the Flask framework. This work was performed using information provided by DGERT within the scope of the REP (Representatividade dos Parceiros Sociais e Impacto da Governança Económica) project that intends to contribute to a more informed decision-making process, more transparent and trustful organizations, and fairer labour relations.

1https://rep.ics.ulisboa.pt/
The rest of this paper is organized as follows: Section 2 presents fundamental concepts and important related work. Section 3 describes the considered solution design. Section 4 presents the results of the developed project. Finally, Section 5 presents the main conclusions, and highlights possible directions for future work.

2 Fundamental Concepts and Related Work

Before initiating the data cleaning process, it is important to evaluate quality of data in a database and then assess its success. The task of evaluating data quality is named data profiling and typically involves gathering several aggregate data statistics which constitute the data profile. It’s used to ensure that the values match up with the expectations (Ganti and Sarma, 2013). Normally, it uses SQL queries to perform the computations. Data profiling can uncover data quality issues in data sources, and what needs to be corrected in ETL (Extract-Transform-Load).

After performing data profiling, it is time to clean the data. Data cleaning is the process of starting with raw data from one or more sources and maintaining reliable quality for applications (Ganti and Sarma, 2013). It consists in a variety of tasks aimed at improving the quality of data. Data cleaning has the following tasks:

- **Record Matching** - match each record from a set of records with records in another table (Ganti and Sarma, 2013);
- **Schema Matching** - align attributes from different schemas (Ganti and Sarma, 2013);
- **Deduplication** - group records in a table such that each group of records represent the same entity (Ganti and Sarma, 2013);
- **Data Standardization** - ensure that all attribute values are "standardized" as per the same conventions (Ganti and Sarma, 2013).

Regarding technological approaches, there are many that enable the development and deployment of effective solutions for data cleaning. The first category consists of Domain-Specific Verticals like Trillium\(^2\) that provides data cleaning functionality for specific domains (Ganti and Sarma, 2013). Since this technological approach understands the domain where the vertical is being used it can tune its solution for the given domain. The second category of approaches relies on horizontal ETL Platforms such as Microsoft SQL Server Integration Services\(^3\) and IBM Websphere Information Integration\(^4\) (Ganti and Sarma, 2013).

These platforms provide a suite of operators including relational operators such as select, project and equi-join. A feature that is common in these frameworks is that applications can plug in their own custom operators. A data transformation and cleaning solution is built by composing the default and custom operators to obtain an operator tree or a graph. The third approach builds upon the extensible ETL platform by extending their repertoire of the default operators beyond traditional relational operators with a few core data cleaning operators such that with much less extra effort and code, a rich variety of efficient and effective data cleaning solutions can be obtained (Ganti and Sarma, 2013).

Regarding generic data cleaning operators, this operators can be used (along with standard relational operators) to build fairly general and accurate data cleaning solutions (Ganti and Sarma, 2013). The most important are:

- **Similarity Join** - responsible for "joining" similar data. It’s very useful in record matching and also deduplication. Two records can be joined if the similarity between their attributes is high. This operation is a fundamental operation to identify approximate duplicate entities in databases and to identify for a given record the best few approximate matches from among a reference set of records;

\(^2\)www.trillium.com  
\(^3\)SSIS - http://msdn.microsoft.com/sql  
\(^4\)https://www.ibm.com/developerworks/data/newto/db2ii-getstarted.html
Clustering - the operation of taking a set of items, and putting them into smaller groups based on "similarity" (Ganti and Sarma, 2013). Clustering is often used in a pre-processing step of deduplication called blocking. When the set of records to be deduplicated is very large, blocking performs a crude clustering to bring similar "blocks" of records together and a finer-grained pairwise comparison is only performed within each block. Another application of clustering is in deduplication itself. Once there are pairwise similarities between pairs of records in a block, clustering based on the pairwise similarities is used to obtain the final deduplicated set of records. In addition to the similarity measure, clustering may be guided by other constraints and by an objective function that determines the best possible clustering among all that satisfy the constraints;

Parsing - the goal of a parsing task is to split an incoming string into segments each of which may be inserted as attribute values at the target.

Regarding data extraction, it consists of filling slots in a database from sub-segments of information. Data Extraction and Data Integration can be performed using different ETL (Extract-Transform-Load) tools, that have specific functionalities useful to achieve this goal. It can also be defined as a family of techniques: segmentation + classification + association + clustering. Segmentation is also known as named entity extraction where we extract from sub-segments of text all the named entities present. Classification consists in separating the entities found into names, companies and positions, e.g. (Microsoft Corporation, CEO, Bill Gates). Different entities can refer to the same thing and association is used in order to separate those who refer to the same thing. Association consists in separating the entities found into associations between the entities found. The last step is clustering where the information is grouped based on the entities that are found (Mooney, 1999).

Data integration can be defined as the set of techniques that enable a uniform access to a set of autonomous and heterogeneous data sources, controlled by different people, through a common schema (Doan et al., 2012). The ultimate goal of data integration is to generate valuable and usable information to help solve problems and gain new insights (Doan et al., 2012). Data integration systems can be warehoused (a data warehouse) or virtual. It’s important to distinguish between virtual data integration systems and materialized data integration systems. Virtual data integration systems are systems where new tables are not created. Instead, when there are two tables and information is needed from both tables, materialized view that uses information of those tables is created. This way a virtual schema - a mediated schema - is created, that will model the kind of answers the users will want when querying the database. Virtual data integration systems also allow getting information from tables in a remote system and a local system. A table that is located in a remote system can be accessed and used when creating a view to be used with a table on the local system (Doan et al., 2012).

Regarding data warehouses, they contain consolidated data from many sources, augmented with summary information and covering a long time period. A distributed DBMS with good scalability and high availability (achieved by storing tables redundantly at more than one site) is required for very large warehouses. An organization’s daily operations access and modify operational databases. Data from these operational databases and other external sources are extracted using external interfaces supported by the underlying DBMS (database management system) (Ramakrishnan et al., 2003). Data warehouses are normally implemented over relational databases or RDBMS (relational database management systems) being designed for query and analysis rather than transaction processing (Kimball and Ross (2013) and Inmon (2005)). It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources. In addition to a relational database, a data warehouse environment includes an extraction, transportation, transformation and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools and other applications that manage the process of gathering data and delivering it to business users. The value of a warehouse is ultimately in the analysis it enables. The data in a warehouse is typically accessed and analyzed using a variety of tools, including OLAP query engines, data mining algorithms, information visualization tools, statistical packages and report generators.
Regarding the web application, it will be developed using the Flask framework. Flask is a web microframework written in Python. Flask has some design principles like the ones described next.

**The Explicit Application Object** A Python web application based on WSGI has to have one central callable object that implements the actual application. In Flask this is an instance of the Flask class. Each Flask application has to create an instance of this class itself and pass it the name of the module.

**The Routing System** Flask uses the Werkzeug routing system which was designed to automatically order routes by complexity. This means that routes can be declared in arbitrary order and will still work as expected. This is a requirement for the users that want to properly implement decorator based routing since decorators could be fired in undefined order when the application is split into multiple modules. The routes try to ensure that URLs are unique, automatically redirecting to a canonical URL if a route is ambiguous.

**One Template Engine** Flask uses the Jinja2 template engine. It is also possible to use a different template engine, although Flask will still configure Jinja2 for the user. Template engines are like programming languages and each of those engines has a certain understanding about how things work. Jinja2 has an extensive filter system, a certain way to do template inheritance, support for reusable blocks (macros) that can be used from inside templates and also from Python code, uses Unicode for all operations, supports iterative template rendering, configurable syntax and more.

**Micro with Dependencies** Flask is a framework that takes advantage of the work already done by Werkzeug to properly interface WSGI (which can be a complex task at times). Thanks to recent developments in the Python package infrastructure, packages with dependencies are no longer an issue and there are very few reasons against having libraries that depend on others.

**Thread Locals** Flask uses thread local objects for request and session objects. Thread locals cause troubles for servers that are not based on the concept of threads and make larger applications harder to maintain. Flask is not designed for those applications since its’ goal is to quickly and easily write a traditional web application.

Regarding information extraction and integration, Zhang et al. (2017) developed a framework named DeepDive that combines database and machine learning ideas to help solve the problems of developing KBC systems. Knowledge base construction (KBC) is the process of populating a knowledge base with facts extracted from unstructured data sources such as text, tabular data expressed in text and in structured forms, and even maps and figures. The authors used an existing knowledge base compiled by human volunteers as a test bed for the KBC research they made. They found that the KBC systems built on DeepDive achieved better quality and accuracy than a knowledge base built by human volunteers over the last decade, being also cheaper and less time-consuming.

Also regarding information extraction, more precisely extracting information from text documents, Govindaraju et al. (2013) reported the importance of extracting information from documents having an approach that joins the inferences done across tabular and text information. Prior approaches used textual and tabular features separately. The authors approach uses both types of features for relation extraction. The authors used standard NLP (Natural Language Processing) features, such as dependency paths, parts of speech and named entity recognition. They believed that a deeper understanding of the text in which a table is embedded will lead to a higher quality table extraction. The authors considered different domains and built for each one of them a system to extract relations from text, tables or both. They concluded that a join inference system that uses standard NLP features can significantly improve the quality of the extracted relations, and that this result holds consistently across all the considered domains.

---

3 Problem Analysis and Solution

In this paper, I describe the project methodology followed, the solution designed for the developed work as also the tasks performed. This work involved applying various data extraction and cleaning techniques to different data sources. Regarding the project methodology used, a "plan, run, test, improve" process was followed. This type of process is common in agile methodology. The agile methodology is based on the concept of ongoing waves or sprints of project planning and execution, enabling to continuously adapt and mature the plan, scope and design throughout the project. This methodology relies on trusting employees and teams to work directly with customers to understand the goals and provide solutions in a fast and incremental way. During the meetings that occurred throughout the development of the project, reviews of progress were made. In my work, I created a database with information about Portuguese Unions and Other Social Partners and developed a web application for manipulation of the database. Figure 1 shows the solution design.

![Solution Design Diagram]

Figure 1: Solution Design.
1 - script responsible for importing the contents of the CSV files containing information about strike warnings into a temporary table.
2 - script responsible for cleaning the information and integrating it into the common database.
3 - script responsible for calling the methods of the web service and importing the obtained information into temporary tables.
4 - script responsible for performing data cleaning of the information and integrating it into the common database.
5 - script responsible for parsing the BTE documents (PDF files) into text files.
6 - script responsible for searching elections mentions in the content of the text files. The mentions found were then stored in corresponding text files.
7 - script responsible for cleaning the information from the mentions text files and integrating the elected managers lists into a common database.

The general procedure is detailed next, through an explanation of all the steps that are involved:

- **Strike Warnings data**: the information was already stored in CSV files. DGERT provided the files and they were saved locally in the project workspace. Then, the files were read using a Python script and their content imported into temporary tables. The Python script was also responsible for cleaning the information and integrating it into a common database;

- **Organizations data**: the information was stored in DGERTs’ database. In order to remotely have access to the information, a web service was developed in collaboration with DGERT. The Python
script makes calls to the methods of the web service (through developed web service PHP scripts) and imports the obtained information into temporary tables. Then, the Python script performs data cleaning of the information from the temporary tables and integrates the information into the common database;

- **BTE documents**: the information was already stored in PDF files. These documents were obtained from DGERT and the Portal do Governo’s website. The PDF files were parsed into text files using a Python script in order to ease the search for the information. Then, a Python script was used that was responsible for reading the text files and searching for the desired information (trade union organizations elections mentions). The elections mentions were stored in text files with the following name format "organizationName_bte[number]_bteYear.txt". These text files were then read in a Python script responsible for cleaning the information from the text files and integrating the elected managers lists into the common database;

- **Web Application**: made throughout the development of the work. It was developed in order to manipulate and search the created database. The web application was developed using Flask framework. The web application consists of a dashboard page and an administration panel. The dashboard has graphic representations about data from the database and an organizations’ search engine that are directly connected (it dynamically updates the graphic representations). The administrative panel allows performing queries and searching the database.

The database defined was a relational database consisting of 12 tables. This database would contain information regarding Portuguese Unions and their activities. Figure 2 shows the conceptual UML model of the developed database.
The database contains data regarding Portuguese unions (Org_Sindical and Org_Patronal tables) and those tables describe all the detailed information about the organizations such as address, phone number and headquarters district. Some tables describe acts made by/between organizations (Outorgantes_Actos, Actos_Eleitorais_Org_Sindical and Actos_Negociacao_Colectiva tables). Regarding information that is contained in BTE documents, tables Mencoes_BTE_Org_Sindical and Mencoes_BTE_Org_Patronal describe mentions to elections and statutes changes referred in bulletins and the organizations that are involved. Tables Direccao_Org_Sindical and Membros_Org_Sindical contain information about some elections that were held for trade union organizations and the elected members. The members table contains personal and confidential information with individual characteristics. The database contains information about strike warnings (Avisos_Greve table) associated to organizations, specifying the organizations involved and some details about the strike such as duration and date.

Regarding the web application, a dashboard page and a administration panel were developed. The dashboard page was developed using flask framework and JavaScript. The representations of data were created using Chart.js module. The administration panel was developed over a SQLite data manipulation tool written in Python. Regarding the dashboard page, there is a login bar for authentication by password in order to concede access to the administration panel, four charts representing statistics about data from the database and a organizations search engine. All the representations allow performing mouse hover, displaying specific information. The organizations search engine by name or acronym was added, with the results found being listed as a table on the dashboard page. By clicking the button to the left of the search
bar, the user can choose if he wants to search for trade union organizations or employers’ organizations. After a search is performed, the four data representations dynamically update regarding the results of the search performed. To enable even more functionalities, a download button was added. This button allows downloading the search results into an excel file. The users of the web application can then use that excel file to analyse and filter the results. Regarding the administration panel, it was used an SQLite data manipulation tool written in Python. There is an index page and different tabs that perform different functionalities. The index page shows some basic information about the database listing all the tables and including the number of tables and indexes, as well as its size on disk. The structure tab displays information about the structure of the selected table, including columns, indexes, and foreign keys (if any exist). From this page you can also create, rename or drop columns and indexes. The content tab displays all the table data. Links in the table header can be used to sort the data. The query tab allows you to execute arbitrary SQL queries on a table. The query results are displayed in a table and can be exported to either JSON or CSV. In order to refresh the information from the database, an option of updating the database was added to this tab. The import tab supports importing CSV and JSON files into a table, that the user can use for research purposes. There is also the option of automatically create columns for any unrecognized keys in the import file.

4 Demonstration

Regarding the developed work, there are two main results:

- The creation of a database describing Portuguese unions and other social partners;
- The development of a web application to be used in order to access and manipulate the created database.

The developed database contains information from various sources like CSV files, PDF files and a relational database. The PDF files contained the bulletins with information about elections and statutes changes. Regarding the BTEs’ information extraction, the information from the Mencoes\_BTE\_Org\_Sindical table was used in order to obtain the bulletin mentions to trade union organizations elections. Table 4.1 shows statistics about the trade union organization BTE elections mentions.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTE elections mentions (1977-2019)</td>
<td>1939</td>
</tr>
<tr>
<td>BTE elections mentions used (2008-2018)</td>
<td>901</td>
</tr>
<tr>
<td>BTE elections mentions that found a match with the organization name</td>
<td>686</td>
</tr>
<tr>
<td>BTE elections mentions that found a match with the elected managers lists</td>
<td>400</td>
</tr>
</tbody>
</table>

Table 1: Statistics about the trade union organizations BTE elections mentions.

Regarding the BTEs’ information extraction, the information from the Mencoes\_BTE\_Org\_Sindical table was used in order to obtain the bulletin mentions to trade union organizations elections. From this table, the trade union organizations elections mentions corresponding to the years range from 2008 to 2018 were selected. The reason for using only this range was because the mentioned bulletins were already parsed into text files and in a format that allowed extracting the elected management bodies lists. This range contained 901 elections mentions. The bulletins described in those mentions were searched in order to find the elected management bodies lists associated to the elections mentioned. From those 901 mentions, 686 found a match with the trade union organization name and 215 did not. Regarding the bulletins where a match was found, they were analysed to see if the match contained the elected management bodies list which was the goal of the extraction. From the 686 mentions that found a match with the trade union organization name, 400 found a match that contained the list of the elected members (corresponding to
58%). These 400 mentions were used and the information was integrated into the Direcao_Org_Sindical and Membros_Org_Sindical tables.

Regarding the developed web application, it consists in a dashboard page and an administration panel. Figure 3 shows the dashboard page. The administration panel is shown in figure 4.

![Dashboard page of the web application.](image)

**Figure 3:** Dashboard page of the web application.
5 Conclusions and Future Work

In this paper, I reported the considered solution design for the work developed. The methodology was discussed as well as the obtained results.

The most important contributions of this work are as follows:

- The data quality is better, since data cleaning techniques were performed;
- A portion of the information from the BTEs PDF documents is now integrated in the created common database making it richer in information. The information on who was elected as a result of the elections held for the positions of union organizations was added to the database;
- The access to the information is now made through the developed Web Application, where the users can make searches, modify and insert new records to tables;

Regarding the work performed, there were some challenges. The pandemic caused many inconveniences while performing this work and some of the tasks had to be done remotely. The Web Service had to be finished working remotely and there were some problems with DGERTs’ remote access that delayed its’ conclusion. Regarding the information about the bulletins, only a portion of the elections mentions was used since there were format problems in a big part of the bulletins and the information extraction using programming would be very difficult to achieve.

 Initially, when the web service was made, the returned information would retrieve all data from each table and this caused an issue. DGERT’s server would frequently go down because of all the information and would be unavailable to all the other DGERT’s services. To fix this problem, each call to the web service would be made for a specific year, reducing the data load on the server. In order to enrich the quality of data, the web service had to be updated several times because during the project meetings was
suggested to have more detailed information about the organizations. The conclusion of the developed web application took a lot of time since it was shown to all the members of the project and improvements were discussed and suggested. Since the dashboard page would be predominantly used by users without experience with SQL, it was important to discuss with the members of the project the most adequate layout. The administration panel is a more technical tool that allows performing SQL queries and needs to be used by an experienced user. Initially, the data representations were static and after meeting with the project members, it was agreed to relate the data representations with the organizations’ search engine. This way, the representations would be dynamic, more interactive and user friendly.

One of the limitations while performing this work were the challenges to extract information from the bulletins (PDF files). The bulletins have different writing formats over the years, which makes it very difficult to extract information programatically. In some cases the boards lists appear in the form of a table, causing problems in the extraction process. Other problem was related to the older bulletins (older than 2008) that weren’t original PDF files and their content couldn’t be parsed to text files using the same method performed for the newer bulletins. For future work, it would be interesting to extract the information from the older bulletins into text files. This could be performed by parsing the PDF files using an OCR (Optical Character Recognition) tool. The dashboard created in the web application has graphic representations of data and a organizations search engine. For future work, the dashboard could have more information like adding more graphic representations and other types of search. The developed database can be used by different institutions in order to perform studies about the social sciences. For future work, it would be interesting to have someone managing the access to the web application in order to allow different institutions to make use of the developed web application in order to manipulate and search the database.
References


