Matching Census Data Records

Rui Menezes da Silva

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Pável Pereira Calado
Prof. Mário Jorge Costa Gaspar da Silva

Examination Committee

Chairperson: Prof. João Emílio Segurado Pavão Martins
Supervisor: Prof. Pável Pereira Calado
Members of the Committee: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

November 2017

Acknowledgments

I would like to thank my fantastic parents for their friendship, encouragement and support over all these years, for always being there in the good and bad times. Without you, I would never come so far. I would also like to thank my exceptional brothers for the companion throughout this experience and to my grandmother for her inspiration and care.

To my amazing girlfriend who supported me since the beginning of this Master and was there when I needed most.

I would also like to acknowledge my dissertation supervisors Prof. Pável Calado and Prof. Mário Gaspar da Silva for their insight, support and sharing of knowledge that has made this Thesis possible.

The colleagues in Statistics Portugal (SP) were fantastic and provide a great insight about the work and also a lot of answers to my questions, namely Sandra Lagarto, Paula Paulino, Anabela Delgado and João Capelo.

To Lufialuiso Velho for the partnership and friendship on this journey in SP.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the good and bad times in my life. Thank you.

To each and every one of you – Thank you.

Abstract

Record Linkage is the task of matching two records that refer to the same entity. This work aims to match records of administrative databases for improving the process of the Portuguese data Census. In this context, this work will help Portuguese Census by reducing the costs and the load over the citizens and provide access to a greater frequency of census information. This dissertation presents methods for record linkage taking into account effectiveness, efficiency and related Census works. Moreover, it proposes an architecture for record linkage based on Supervised Learning as well as methods to evaluate the results.

Keywords

Record Linkage, Census, Machine Learning, Logistic Regression

Resumo

O processo de emparelhar dois registos, que se referem à mesma entidade, é denominado por Emparelhamentos de Registos (Record Linkage). Este trabalho tem como objetivo emparelhar registos de bases de dados administrativas para melhorar os Census em Portugal. Desta forma, este trabalho irá ajudar a reduzir os custos nos Census, reduzir os inquéritos sobre as pessoas e possibilitar uma maior frequência aos dados censitários. Além disso, esta dissertação apresenta métodos de Emparelhamento de Registos tendo em conta a eficácia, eficiência e trabalhos relacionados com os Census. Também, será apresentado a arquitetura da solução baseada em Aprendizagem Supervisionada, assim como métodos para avaliar os resultados.

Palavras Chave

Emparelhamento de Registos, Census, Machine Learning, Regressão Logística.

Contents

List of Figures

List of Tables

Acronyms

SP Statistics Portugal

IST Instituto Superior Técnico

BPR Resident Population Base

BDIC Civil Population Register

IISS Informatics of Social Security Institute

AT Tax Authority

CGA General Retirement Fund

EDUC General Statistics of Education and Science

IEFP Unemployment and Vocational Training Institute

SEF Immigration and Borders Service

CNPD Data Protection Authority

NIC Civil Register Identifier Number

NISS Social Security Identifier Number

NIF Finances Identifier Number

AR Residence Authorization Number

SVM Support Vector Machines

SNM Sorted Neighbourhood Method

IA-SNM Incrementally-Adaptive Sorted Neighbourhood Method

AA-SNM Accumulatively-Adaptive Sorted Neighbourhood Method

RR Reduction Ratio

1

Introduction

Contents

Every ten years in Portugal, and in so many other countries, Census are performed. Census is the biggest statistical operations in every country. So far, in Portugal, Census is performed based on the Traditional Model, i.e, as a door to door questionnaire. However, some countries began to use an Administrative Model, where the Census data are progressively obtained from Public Administration databases.

The Administrative Model carries some advantages, like reducing the costs and the load over the citizens, and providing access to a greater frequency of census information. It also has a better effectiveness on statistical production and covers the lack of information of the Traditional Model. For example, in Portugal, the last Census cost over 45.2 million euros, showing that there is an opportunity to reduce the cost by collecting data from databases.

In this context, Statistics Portugal (SP), the entity responsible for the Census, started a feasibility study to transform the data collection process of the Portuguese Census to a hybrid model (a combination of the Traditional and the Administrative Model) [1]. Currently, SP has access to more that 10 administrative databases. The administrative databases used in this work are:

- Civil Population Register (BDIC)
- Tax Authority (AT) (IRS)
- Informatics of Social Security Institute (IISS)
- General Statistics of Education and Science (EDUC)
- General Retirement Fund (CGA)
- Unemployment and Vocational Training Institute (IEFP)
- Immigration and Borders Service (SEF)

In order to perform statistical analysis, SP must link/match the records between the databases (those that refer to the same person). Thus, SP is able to apply some residence rules to estimate the number of resident people in Portugal in a determined year with the creation of a database, named the Resident Population Base (BPR). BPR contains the records, matched from the different administrative databases, of resident people with the corresponding data. Table 1.1 shows a sample of the BPR.

Table 1.1: Example of BPR where NIC, NIF and NISS correspond to different personal identifiers/keys of different databases

NIC	NIF	NISS	NAME	DATE OF BIRTH	SEX	BIRTH COUNTRY	POSTAL CODE
F34F3F43	-	-	RUI LVA	-	M	PORTUGAL	-
C34V3V3	2BBYU6RE	-	MAR CIA	20-12-1995	-	-	-
FE24V43	KNUD9S9S	S0EDWC9C	MAN ZES	11-01-1989	M	PORTUGAL	1590-741
-	-	B65H654S3	TOM NIO	-	-	-	-
-	D4D3FF33	-	JOA GAS	-	F	BRAZIL	-

Every database has one or more personal identifier/key that identifies the record like Civil Register Identifier Number (NIC), Finances Identifier Number (NIF), Social Security Identifier Number (NISS) or Residence Authorization Number (AR). One way of matching records is with a common key. However, when it was not possible to match the records through the key, SP uses exact methods to match the records. For instance, if the names are equal and if the dates of birth are equal and if the nationalities are equal, then it is the same person.

Table 1.2 illustrates why it is required to use record linkage techniques. If all the databases have at least one personal key in common, with total completeness, it would be possible to join the databases and match the records, however, that does not happen and alternative methods are needed.

Table 1.2: Completeness of the Personal Identifiers/Keys for each Database (in percentage)

	PERSONAL KEYS			
DATA SOURCES	**NIC**	**NIF**	**NISS**	**AR**
BDIC 2015	100	-	-	-
AT 2015	-	100	-	-
IISS 2015	81.5	97.8	100	-
EDUC 2015	92.5	-	66.6	5.5
IEFP 2015	100	99.6	98.8	-
CGA 2015	79.8	86.1	-	-
SEF 2015	-	62.2	50.8	100

1.1 Challenges

There are multiple hurdles to the creation of the BPR:

1. Each administrative database has millions of records to match.

2. The Portuguese Constitution prevents the State from assigning a single unique number to citizens, so each information source has a different key.

3. Individuals may be only partially registered or not even registered in some of the data sources.

4. Data Protection Authority (CNPD) imposes the anonymisation and pseudonymisation of the datasets provided to Statistics Portugal.

5. Records have inconsistencies, errors and different representations due to manually inserted data.

The first problem hinders this task because of the number of comparisons needed when performing record linkage between two databases. For instance, if we have two databases with one million records each, if we compare all the records of one with all the records of the other it would lead to over a trillion of comparisons. All these comparisons are unfeasible, therefore it is necessary to find a method to reduce the number of comparisons.

Following, it is not possible to have the same common personal identifier across all datasets. Nevertheless, some datasets share the same personal identifier. Table 1.2 shows the databases and the respective personal identifiers, that in fact are keys. The main database for the Portuguese population is the BDIC because Portuguese citizens have to register there. So, for instance, if we try to match through key BDIC and IISS we can use the NIC key, but if we try to match BDIC and AT we have to use IISS as an intermediate because this database contains both NIC and NIF.

The records may not have all the fields filled, as it happens on the key field. In this case, we needed to use a different matching method. SP used exact methods but some records have few fields in common to compare and some of them are even null.

Matching through keys could match possibly all the records of most databases. Unfortunately that does not happen, even when both keys are filled for every record, because: the records from each database could have a time difference caused by the time it was transferred to SP that alters some important data. Also, many records can be outdated. For example, a person may die and only one database is updated, or a nationality change is registered in only one database. Finally, there is the hypothesis of errors on the records, even in the key field.

Furthermore, the anonymisation and pseudonymisation imposed by the CNPD raise the difficulty of this problem due to the following impositions on data provided to SP:

- Pseudonymisation of the personal identifier though encrypted hash.

- Access only to the first three letters of the first name and the last three of the last name.

- Absence of the address.

These restrictions are illustrated on Table 1.3.

Table 1.3: Example of the Database Social Security with the restrictions imposed by CNPD

Social Security		
NISS	**NAME**	**ADDRESS**
EFD8W4E8F8WE8	MAR ZES	-
C6WESD5CWE84D	JES TAS	-
FEW49R81VSZWF	SUS ROS	-

Pseudonymisation does not really affect the task because it is encrypted with the same encryption method for all keys of all databases. On the other hand, the anonymisation, by truncation of name makes this work more challenging. The complete name is an attribute that distinguishes people. Reducing it to three letters the first name makes it hard to match, especially in the cases of common first names like ”Maria” or ”José”.

Last but not least, the errors on the data also cause a problem in the linkage process. In Table 1.4, we show some examples of the errors in the data. Each pair of records from each database corresponds to the same person. The first pair has an error in the last name. Moreover, the second pair has a different name for the same city and finally, the last pair has the day and month switched. These errors and inconsistencies hinder the record linkage and are one of the main causes for SP not find all true matches across the databases with exact methods.

Table 1.4: Example of the errors of records from two different databases

Records from Database 1			**Records from Database 2**		
Name	**City**	**Date of Birth**	**Name**	**City**	**Date of Birth**
RUI ZES	HORTA	14-06-1994	RUI SES	HORTA	14-06-1994
MAR RRA	PORTO	25-04-1959	MAR RRA	OPORTO	25-04-1959
JOA NIS	COIMBRA	09-10-1999	JOA NIS	COIMBRA	10-09-1999

1.2 Goals and Contributions

The goal of this work was to design a record matching model, that will receive records from different databases and determine if the records are or not duplicates, in other words, find all records that refer to each person. This will make possible to SP to know which record in Civil Population Register (BDIC) corresponds to another in Tax Authority (AT), for example. With the discovery of new matches, SP will be able to apply the residence rules and check if the person from the linked records is a resident or not and also fill the gaps in the BPR by adding new records and consequently new attributes.

The record matching model uses probabilistic methods. This matching model has a component of monitoring as well.

SP already started the creation of the BPR. As a consequence of the problems referred in Section 1.1, they were unable to match them all. Table 1.5, represents the number of records of each database, the number of records SP matched into BPR and the last column is the number of records that were not possible to integrate into BPR. The number of records which are not in the BPR for IEFP are negative because SP has a problem of duplicates, by that time, leading to matching extra records. In this perspective, our job is to help SP find new matches in order to increase the number of connected records in BPR and complete the null fields with the respective accurate data, allowing SP to answer one very important question – How many people live in Portugal?

This work resulted in an article for the Data Science, Statistics and Visualization conference [2].

Table 1.5: Table showing the numbers of records for each database and the records that are not integrated in BPR.

DATABASE	NUMBER OF RECORDS	NUMBER OF RECORDS INTEGRATED IN BPR	NUMBER OF RECORDS NOT INTEGRATED IN BPR
BDIC 2015	11 825 786	9 985 188	1 840 598
AT 2015	9 370 879	8 969 050	401 829
IISS 2015	6 927 720	6 678 767	248 953
EDUC 2015	1 777 732	1 667 252	110 480
CGA 2015	1 032 133	1 001 865	30 268
IEFP 2015	746 855	752 336	-5 481
SEF 2015	383 759	218 814	164 945

1.3 Methodology

This work started with a partnership between Instituto Superior Técnico (IST) and SP, where I and a colleague from IST – Lufialuiso Sampaio Velho – both worked towards the goal of automating the Portuguese Census procedure [3].

Our methodology started with an understanding of the problem and the environment of work with SP staff. We signed a confidential agreement in the beginning due to the access of sensitive data. We start by accessing the data to learn how the databases were organized and if we have the necessary privileges to perform queries and create tables, for example. It was also necessary to install the programming language Python and the scikit learn and levenshtein libraries.

At the same time, we planned an architecture based on probabilistic methods, since exact methods would not add any value to SP. We approached this work as a classification problem. We opted for a logistic regression classifier because it has the requirements to solve the problem [4]. Further, we used one the most commonly used candidate selection methods – Standard Blocking [5] – to reduce the number of comparisons. In the first test, we did a 2 fold cross-validation with BDIC and AT with 98% of precision and recall. For that reason we kept using this method to match records, with some tuning over time. In parallel, we did the Monitoring module to evaluate the quality of the process, from the beginning to the end.

1.4 Organization of the Document

This work is divided into 6 Chapters. Chapter 1 introduces to the problem, the methodology used and the expected goals for this work. Chapter 2 presents some concepts of methods used in Record Linkage, while Chapter 3 shows various works related with this problem in terms of efficiency, effectiveness and also Census works. Chapter 4 explains the architecture of the project, as well as the Monitoring Module. Chapter 5 analyses the results achieved. Finally, Chapter 6 has a reflection about the work and some thoughts about the improvements for future work.

2

Background

Contents

Record Linkage is the task of finding records in different databases that refer to the same entity, even if the records are not identical [5]. It is commonly used for *”improving data quality and integrity, to allow reuse of existing data sources for new studies, and to reduce costs and efforts in data acquisition”* [6].

In Fig. 2.1, we can observe a schema that illustrates a Record Linkage process. In summary, Record Linkage starts with **Data Cleaning and Standardisation**. **Data cleaning** is the process of replacing, modifying or deleting dirty data (incorrect or inconsistent data) in order to have reliable data and avoid errors. **Standardisation** or **Normalization** is the process of having the data in the same consistent format across all databases in the way that has the same representation.

Indexing refers to a candidate selection of the records, in other words, select which records will be paired to be compared afterwards because we cannot compare them all, as was explained in Section 1.1.

In the step **Record Pair Comparison**, the previous selected records are compared using similarity metrics (see Section 2.3).

Similarity Vector Classification uses the similarity scores and classifies the records as Matches, Non-matches or Possible matches, where the last, could be manually labeled by an experient user or expert as Match or Non-Match on the step **Clerical Review**.

Additionally, in module **Evaluation** we can evaluate the retrieved results so that is possible to adjust some parameters and check if the process is working as expected.

In the following sections, some of the basic concepts on Record Linkage are presented.

2.1 Blocking

One of the techniques usually applied to speed up record matching is *Standard Blocking* [5]. Standard Blocking is an indexing technique that consists of grouping records that are similar by using a blocking key. In Figure 2.2, I exemplify how Standard Blocking works when applied to a single table. In this example, the blocking criteria (the rule that determines how the blocking key is formed) for the blocking key is the concatenation of the attributes last name and postal code. In this perspective, the first record with the last name *Silva* and the postal code *9900-222* will have the following blocking key - *Silva9900222*.

In Figure 2.2 it is the table with all the blocking keys from the table above. Standard Blocking groups the records that have the same blocking key in a block. In this example, only two records have the same blocking key (*Pereira1350358*) and consequently will go to the same block. The premise is that only the records that are in the same block will be compared with each other. In this respect, we reduce the number of comparisons from six to one. Furthermore, if we imagine a table with millions of records, it is easy to see the potential reduction of the number of comparisons after Standard Blocking eliminates for the search space the records with keys outside each block. The reduction of comparisons is directly

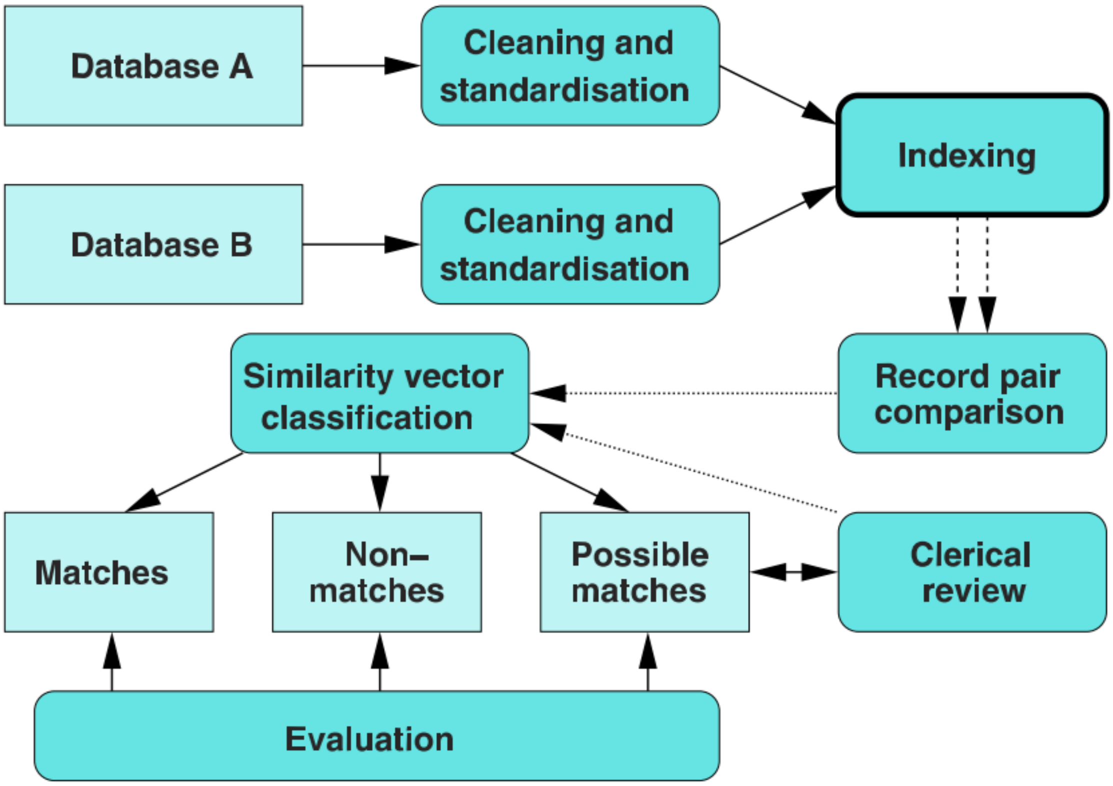


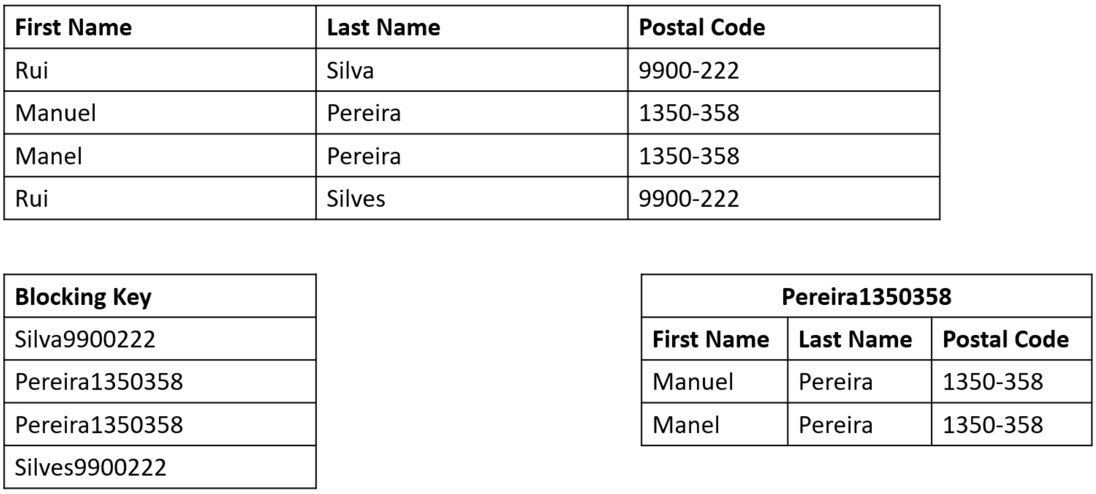
Figure 2.1: Schema of Record Linkage Process, taken from [6]

affected by the blocking criteria. For instance, if we try to match persons and use as blocking criteria – First Name + Date of Birth + Nationality – it will have a higher reduction on the number of comparisons when compared with the blocking criteria – sex. The trade-off of using this indexing technique is between reduction on comparisons and false negatives. The more we reduce the number of comparisons, more potential false negatives we could have.

Another approach is the Sorted Neighbourhood Method (SNM) where the records are sorted according to the blocking key [7]. Figure 2.3, represents the table with the blocking keys from the example from Figure 2.2. The goal is to move a window of a fixed size (bigger than one) through the ordered blocking keys. The records whose blocking keys are covered for the window will go to the same block. Similar to **Standard Blocking**, records will be compared pair-wise only with the ones in the same block.

2.2 Machine Learning Techniques

One approach to record linkage is to classify pairs of records as being a match or not. To this effect, we can use several types of machine learning models. These models can be obtained through **Supervised Learning**, **Semi-Supervised Learning**, **Active Learning** or **Unsupervised Learning**.

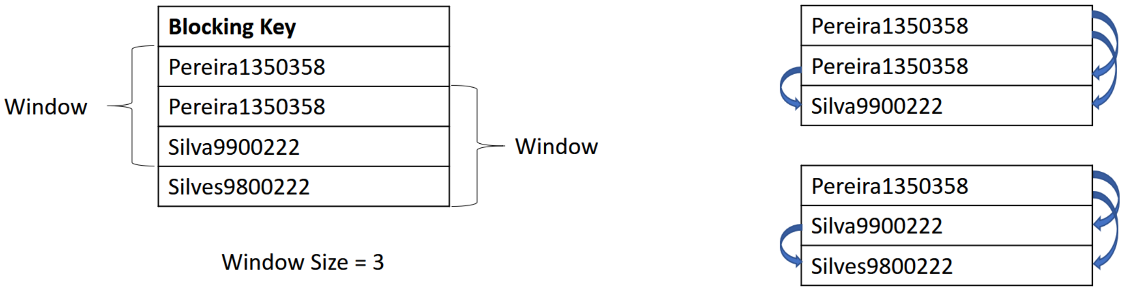


First Name	Last Name	Postal Code
Rui	Silva	9900-222
Manuel	Pereira	1350-358
Manel	Pereira	1350-358
Rui	Silves	9900-222

Blocking Key
Silva9900222
Pereira1350358
Pereira1350358
Silves9900222

Pereira1350358		
First Name	Last Name	Postal Code
Manuel	Pereira	1350-358
Manel	Pereira	1350-358

Figure 2.2: Example of Standard Blocking



Blocking Key
Pereira1350358
Pereira1350358
Silva9900222
Silves9800222

Pereira1350358
Pereira1350358
Silva9900222

Pereira1350358
Silva9900222
Silves9800222

Figure 2.3: Example of Sorted Neighbourhood Method, with a window size 3

In **Supervised Learning** is given a training set, in which the data is labeled [8]. In the context of this work, the training set would be pairs of records labeled as match or non-match. This training set would be submitted to a supervised algorithm to create a model. Therefore it is possible to apply the model to unlabelled data and mark as match or non-match. Possible algorithms are Support Vector Machines (SVM) [9], Decision Trees [10] and Logistic Regression [4], among others.

On the other hand, with **Semi-Supervised Learning**, the training data contains not only labeled data but also unlabeled data, since it is easier and cheaper to acquire unlabeled data [11]. It uses labeled data to build a classifier and applies it to the unlabeled data. Examples of semi-supervised algorithms are self-training [12] and co-training [13].

Active Learning is a learning technique that is based on an initial small training data set of labeled instances [14]. The learning algorithm first trains with the labeled data set and then asks queries to an experienced user for labels, (see Figure 2.4). Since the learning algorithm chooses the best examples, it learns faster and with less labeled data than regular supervised learning. For example, these authors studied the active learning for SVM [15].

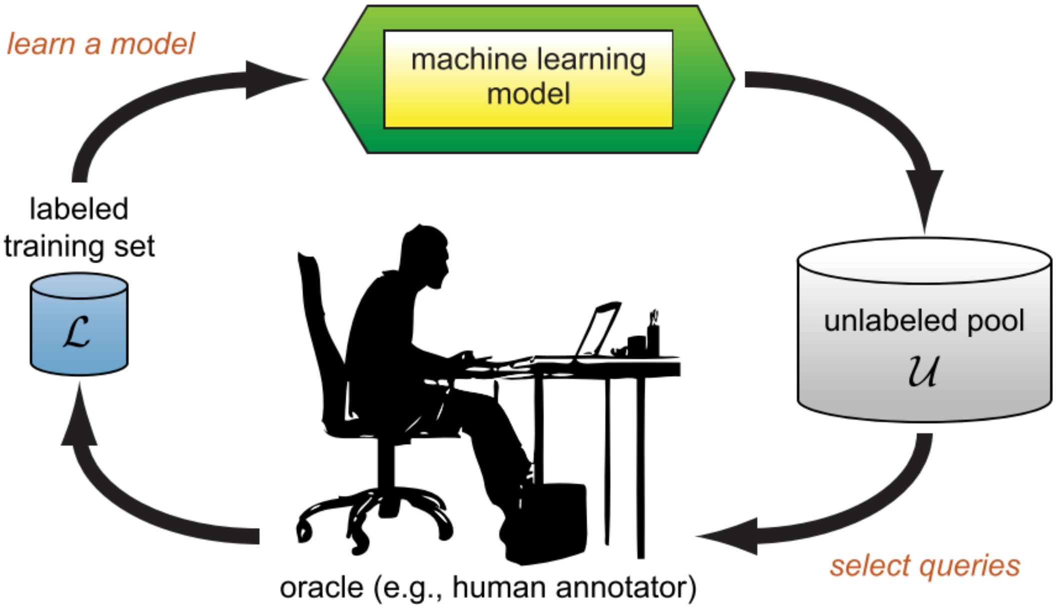


Figure 2.4: Active Learning Cycle Example taken from [14]

Unsupervised Learning, on the contrary, uses only unlabeled data [8]. It tries to find patterns in the data and group it, to form clusters which could represent a class. Examples of methods are K-Means [16] and Self Organized Maps [17].

The algorithm logistic regression is a supervised learning method used usually in binary classification (classification with only two possible outcomes). This method uses the logistic function, also named sigmoid function (see Equation 2.1), to calculate the probability of a given attribute or set of attributes corresponds to a defined class, normally set as 0 or 1.

$$\sigma(t) = \frac{1}{1+e^{-t}} \tag{2.1}$$

The input values, t, of the logistic function, are combined linearly using weights, normally represented as β. Equation 2.2 represents the same logistic function with the weights β_1 and β_2 over the input t.

$$\sigma(t) = \frac{1}{1+e^{-(\beta_1+\beta_2\times t)}} \tag{2.2}$$

These weights or coefficients can be calculated with maximum-likelihood estimation using training data. Consider the example of the factors of a political candidate wins an election or not. To keep this example simple there will be just one factor considered – the amount of money spent on the campaign (t). Thus, we need a sample of examples to train in order to calculate the weights for the model. Afterwards, it is possible to determine the probability. As an example, imagine that β_1 is -4 and β_2 is 1.5. The probability of candidate win the election knowing that spent 5 thousand euros in the campaign is $\sigma(5) = \frac{1}{1+e^{-(-4+1.5\times5)}} = 0.9706$. As a result, the logistic regression predicts that the candidate will win the election because the returned value is closer to 1 than 0.

2.3 String Similarity Metrics

This Section introduces some of the most commonly used similarity metrics regularly used nowadays. The goal of using each one of these metrics is to have a value that describes how much two strings are alike. These metrics are important in Record Linkage because the fields of records usually have typographical errors. For this reason, we have to compare the fields with a metric that returns a score of similarity. Thus, we know how much two strings are alike or not.

The Edit Distance between two strings is the minimum number of operations (insertions, deletions, and substitutions) to transform a string into other. The most famous edit distance was proposed by Levenshtein and each operation has a cost of one [18]. The formula is represented on the Equation 2.3.

$$d(i,j) = min \begin{cases} d(i-1,j-1) + c(x_i,y_j) & \text{copy or substitute} \\ d(i-1,j) & \text{delete } x_i \\ d(i,j-1) & \text{insert } y_j \end{cases} \tag{2.3}$$

c(x_i,y_j) = 0 if x_i = y_j, 1 otherwise

d(0,0) = 0; d(i,0) = i; d(0,j) =j

Table 2.1 illustrates the similarity score calculated with Edit Distance between two strings. The similarity between Jonh and Jon is one because it is necessary only one operation to transform a string into the other. The operation is the deletion of h in Jonh to transform in Jon.

Table 2.1: Example of Edit Distance between the strings: Jonh and Jon

		J	**O**	**N**	**H**
	0	1	2	3	4
J	1	0	1	2	3
O	2	1	0	1	2
N	3	2	1	0	**1**

The complexity of an edit distance between s_1 and s_2 is O($|s_1| \times |s_2|$).

Q-grams are a contiguous sequence of q characters of a string. To get, for instance, a bigram (q-gram of size 2), a window of size 2 slides over the string and the characters covered by the window form bigrams. For example, the bigrams of the word ”paper” are [pa],[ap],[pe],[er].

The similarity between string s_1 and s_2 is calculated using the Jaccard coefficient, presented on

Equation 2.4, where $|s_1 \cap s_2|$ is the number of common q-grams of s_1 and s_2 and $|s_1 \cup s_2|$ is the size of union of s_1 and s_2 q-grams.

$$Jaccard(s_1, s_2) = \frac{|s_1 \cap s_2|}{|s_1 \cup s_2|} \tag{2.4}$$

The Jaro metric was developed primarily to compare first and last names. The formula is presented next:

$$Jaro(s_1, s_2) = \frac{1}{3}\left(\frac{c}{|s_1|} + \frac{c}{|s_2|} + \frac{c-t}{c}\right) \tag{2.5}$$

Where $|s_1|$ and $|s_2|$ and are the lengths of the strings s_1 and s_2, respectively. *c* is the number of common characters where s_1[i] = s_2[j] and $|i - j| \leq min(|s_1|, |s_2|)$. Finally, t is the number of transpositions, comparing the *ith* character of s_1 with the *ith* character of s_2. If they are different there is a transposition.

Jaro-Winkler is a modification of Jaro based on the fact that fewer errors appear at the beginning of names [19].

This metric adds two new parameters, PL and PW, where the first is the length of the longest common prefix between the two strings, and the second is a given weight of the prefix. The formula is presented next:

$$Jaro - Winkler(s_1, s_2) = (1 - PL \times PW) \times jaro(s_1, s_2) + PL \times PW \tag{2.6}$$

The metrics so far were presented were character based. Soundex is based on the phonetic representation of the string, commonly used for matching names. For a given string the metric works like this:

- Keep the first letter of the name
- Ignore any occurrences of the letter W and H
- Map the remaining letters with the following codes:
  - B, F, P, V with 1
  - C, G, J, K, Q, S, X, Z with 2
  - D, T with 3
  - L with 4
  - M, N with 5
  - R with 6
- The vowels, A,E,I,O,U and Y are not replaced

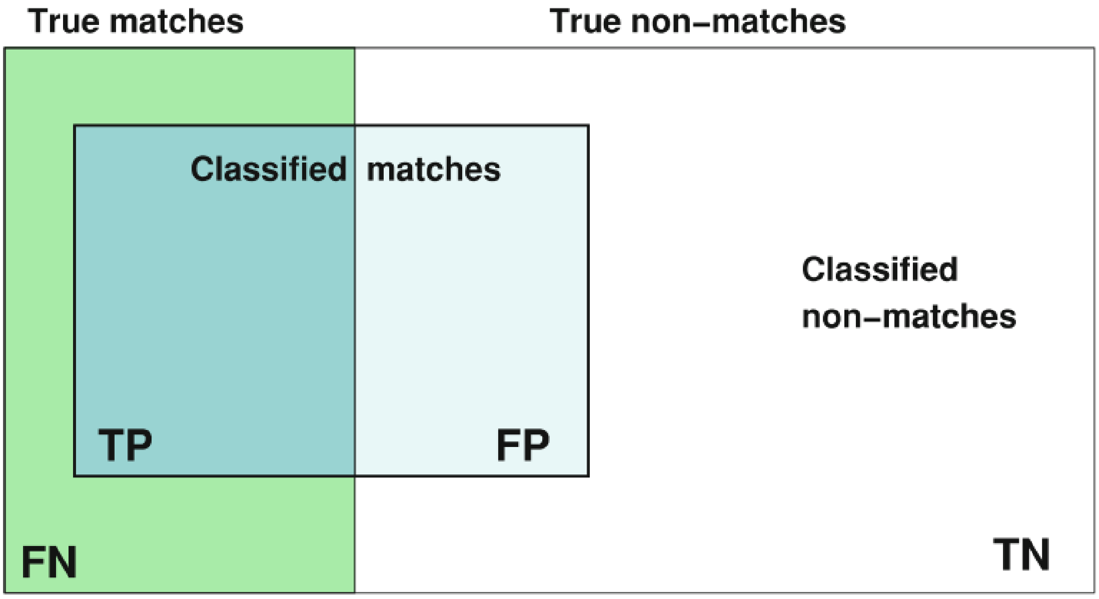


Figure 2.5: Example of the Outcome of the Classification, taken from [6]

- Merge sequences of the same digit into the digit itself
- Drop the vowels and Y, except if it is the first letter
- Keep the first 4 letters, if it has less than 4 characters, add zeros

For instance, the names Michael and Mikael are similar when pronounced and for that reason have the same output when submitted to Soundex – M240.

2.4 Evaluation Metrics

A good evaluation is important to check if the solution has the quality required for this work.

Figure 2.4 shows the possible outcome for every result. True Positives (TP) are the data classified as match when it is a true match in the gold standard. The gold standard is the set that contains the desired results. In this work, the gold standard, are the records matched through the common key because these matches we have sure that they are correct. For this reason, we apply some of the following metrics on the matched records using a 2-fold cross validation. Further, True Negatives (TN) are the data classified as non-match and is a true non-match in the gold standard. In contrast, the False Positives (FP) are the data classified as match and is a non-match in the gold standard and False Negatives (FN) are the data classified as non-match and is a true match in the gold standard.

Using these values, it is described some metrics used for works of Record Linkage.

Precision is the ratio of True Positives that are in the gold standard. It is presented in the following equation:

$$Precision = \frac{TP}{TP + FP} \tag{2.7}$$

In this work context, how many classified matches are, in fact, matches.

Recall represents how many matches (according to the gold standard), are classified as matches and it is presented in the following equation:

$$Recall = \frac{TP}{TP + FN} \tag{2.8}$$

F-Measure is the harmonic mean of **Precision** and **Recall** to find an intermediate value between these two metrics. It only has a high value if both **Precision** and **Recall** have it. **F-Measure** is described in the following equation:

$$F - Measure = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{2.9}$$

2.5 Summary

This Chapter explained how the Record Linkage process is performed. It is crucial to understand this process because the goal of this work is to pair records from different databases.

In this work, for the Indexing step, we opted for the Standard Blocking instead of SNM because it is faster (it is not necessary to order the blocking keys), every record with the same blocking key will be compared and similar Census works use it and have proven results.

Supervised Learning was the obvious choice because we have many labeled data available. Among the Supervised methods, we chose logistic regression because is very good in binary classification and is also very fast.

Since most of the fields have small strings or are codes we opted to use for every fields the Edit Distance and the results were good.

Finally, these evaluation metrics are important to control the quality of the process and are commonly used in classification problems.

3

Related Work

Contents

This Chapter presents some works focused on accomplishing efficiency or effectiveness in Record Linkage. As we saw in Fig.2.1, the Indexing step is where we can achieve a greater efficiency and in the Record Pair Comparison and similarity vector comparison steps the better effectiveness.

Since this is the focus of our work, we end this Section by describing some works applied to the problem of Census data.

3.1 Duplicate Detection Efficiency

Efficiency is a matter of great importance in duplicate detection. Yan et al. show two different approaches of the SNM [20]. The first algorithm, Incrementally-Adaptive Sorted Neighbourhood Method (IA-SNM), basically tries to adjust the window size if the records are similar or not. Instead of the window size being constant, it grows or shrinks, if the distance between the first and last record of the window is below or above a given threshold. As a result, similar records will be in the same block and therefore will be compared, while the less similar will be in different blocks and will not be compared.

The second algorithm, Accumulatively-Adaptive Sorted Neighbourhood Method (AA-SNM), tries to find the boundary pairs (adjacent record of the first record of the window that has a distance above a given threshold) as quick as possible, compared with IA-SNM by creating consecutive larger windows. When it finds the boundary pair in the last window, that will be the largest, it does the same thing as before, but instead of creating consecutive larger windows, creates smaller sub-windows to find the boundary pair in order to set the end of the window. Then it groups the previous adjacent windows into blocks by transitivity.

Figure 3.1 shows comparisons between the different algorithms. Despite the Reduction Ratio (RR) for IA-SNM and AA-SNM is lower compared with Exact Blocking and the SNM, the F-measure, that is the harmonic mean between RR and pairs completeness (PP), reveal that both IA-SNM and AA-SNM outperform the others.

McCallum proposes the use of canopies [21]. Canopies are similar to clusters, the difference being that they are created with a cheap similarity measure and overlap each other. Afterwards, a better and more expensive similarity measure is applied between the records of the same canopies. In this perspective, the data points, or in this case records, that are in separate canopies will be sufficiently different from the others in different canopies. Because the similarity measure is cheap and the canopies overlap, duplicate records will probably be compared.

Monge and Elkan proposed an algorithm that works through transitive closure, that is, if *a* is duplicate of *b* and *b* is a duplicate of *c* then *a* is a duplicate of *c* as well [22]. The structure used was an undirected graph in which the nodes are records and the edges between the nodes represent if they are duplicates.

Bigram indexing is a method, which essentially transforms each blocking key value in a bigram and

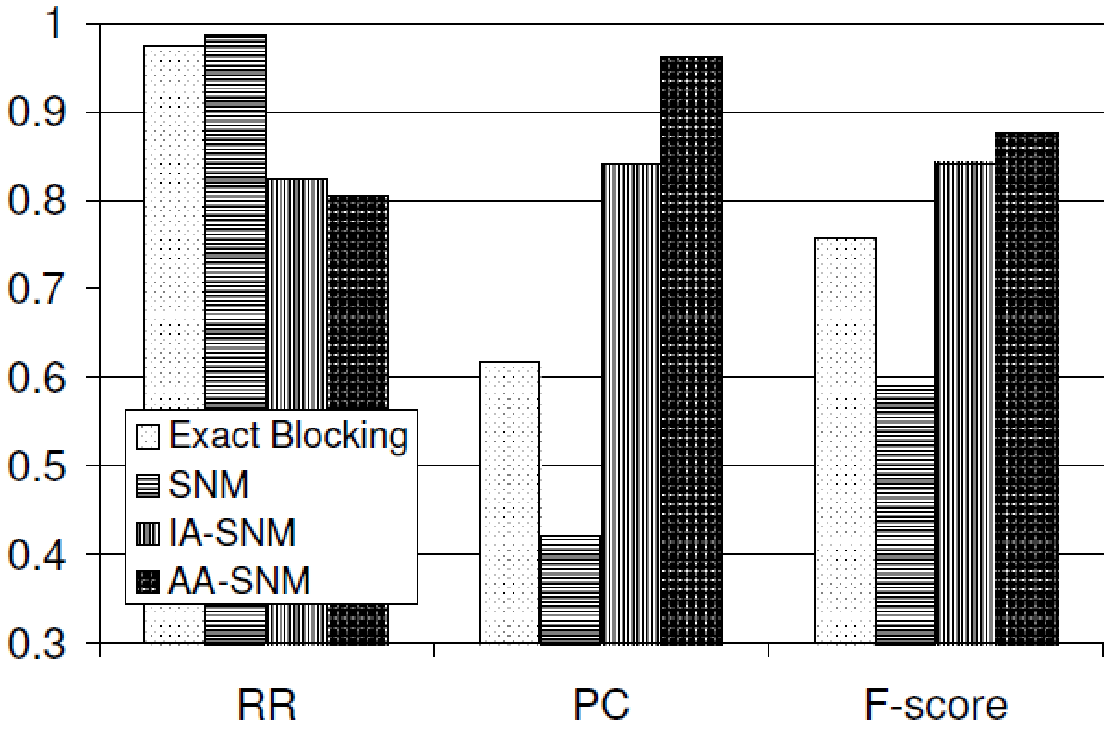


Figure 3.1: Comparisons between the Different Algorithms on the Cora Dataset

then a threshold (with values between 0.0 and 1.0) is applied to form sub-lists of permutations [23]. For instance for the blocking key value *window* the bigrams are 'wi' 'in' 'nd' 'do' 'ow'. If the threshold is 0.8, to calculate the bigrams sub-lists, we multiply the threshold times the length of the list of bigrams $(5 \times 0.8 = 4)$. As a result, we get the sub-lists of length 4 with all the possible permutations that afterwards will be inserted in an inverted index. Evaluations of this algorithm show a trade-off: while bigrams retrieve more true matches it has to do more comparisons between the bigrams.

Cochinwala et al. approach this problem with the reduction of complexity of the Machine Learning rules by pruning some of the fields of the records [24]. So there is a trade-off between complexity and classification accuracy.

A different approach is proposed by Jin et al. [25], where the blocking key values are converted to a multidimensional Euclidean space, using a function named StringMap, a modification of FastMap [26]. Then a multidimensional similarity join is applied to determine similar pairs of records to form clusters where, at last, the records will be compared with a similarity metric.

3.2 Duplicate Detection Effectiveness

This Section presents some relevant techniques in the process of record linkage.

The problem of comparing dates is addressed in [6], where these are compared based on the difference of days using numeric absolute difference presented in Eq. 3.1. The variable d_{max} represents a threshold for the maximum of days that the difference between the days d_1 and d_2 could have.

$$sim_{day\_abs}(d_1, d_2) = \begin{cases} 1.0 - (\frac{|d_1 - d_2|}{d_{max}}) & \text{if } |d_1 - d_2| < d_{max} \\ 0.0 & \text{else} \end{cases} \tag{3.1}$$

Sarawagi et al. presented ALIAS [27], which is a system of duplicate detection that uses *Active Learning*. The premise is that if the learning method chooses from the unlabelled instances those that are more uncertain it will improve and strengthen the classifier at the fastest possible rate. To discover these most uncertain instances is used a committee of classifiers, different from each other, but with similar accuracy. The data that is assigned different labels from the classifiers are the uncertain ones.

We can use Threshold-Based-Classification to classify the records into matches, non-matches and potential matches [28]. One basic way is to sum all similarity scores of the fields, previously compared between two records with a similarity metric, and if it is above an upper threshold it is a match and if it is below a lower threshold, it is a non-match. If it is in the middle of the thresholds, is a potential match and needs clerical review by a specialist or experienced user.

Some attributes are more important to compare records. For instance, the sex of a person is less distinct between records, than the date of birth. So instead of just summing the similarity scores, a weight could be applied to each similarity score of a particular field. Fields like first name, last name and date of birth would have a higher weight than sex or nationality, for example.

Another approach is Rule-based methods where experts with a high domain knowledge of the database create a set of hand-crafted rules to be applied to the results of the similarity scores [29]. An example rule could be:

$s(Surname) > 0.75 \wedge s(DateofBirth) = 1.0 \implies Match$

where s(*field*) stands for a similarity function that is applied to the fields of two records. It is possible that rules classify into matches, potential matches and non-matches.

Rule-based approaches can reach a very high accuracy, but require a lot of tuning, a high knowledge of the data set by the expert, being a very complex task. As a consequence, machine learning is commonly used to create a model and afterwards, the generated rules are tuned.

Galhardas et al. present a data cleaning framework implemented as a data flow graph of data transformations where each node represents an operation, applied over an SQL query [30]. These operations are: *mapping*, *matching*, *clustering*, *merging* and *view*. The *mapping* operator gives a new structure and standardizes the input data. The *matching* operator applies a similarity metric over two relations. Furthermore, the *clustering* operator groups the data set, previously generated by the *matching* operator, in clusters based on the similarity scores results computed. Afterwards, the *merging* operator merges the clusters into a single tuple. Finally, the *view* operator allows to check the integrity of an SQL result.

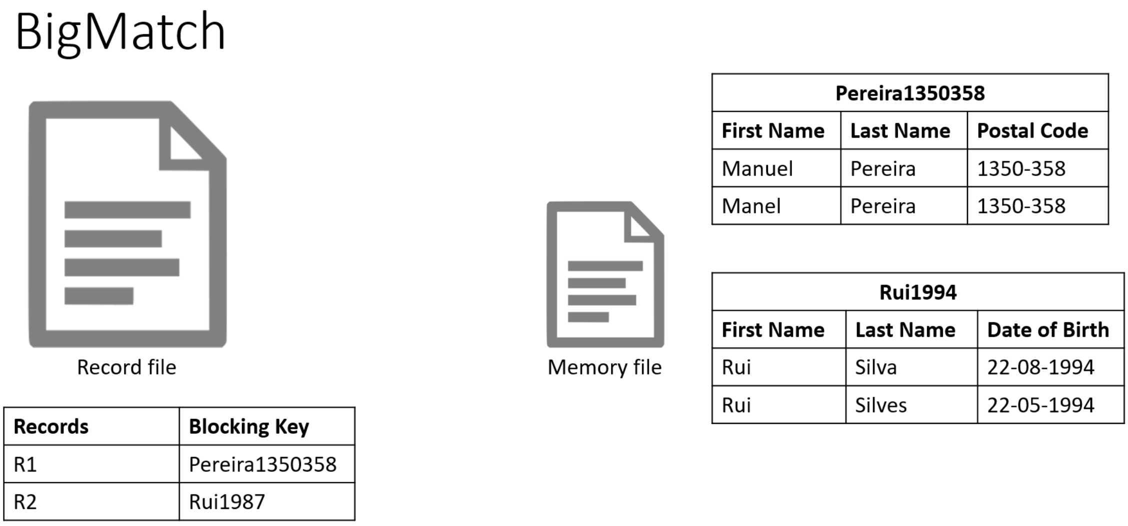
Elfeki et al. present a record linkage toolbox – TAILOR [31] – and propose two models using decision trees, comparing them with the probabilistic record linkage model. In the first, the training data is manually labeled by an expert and afterwards, it is trained by the classifier. The second one is a mixture of *Supervised* and *Unsupervised Learning*, named by the authors, Hybrid Record Linkage Model. The fact that labeled data is difficult and exhausting to manually label, we can use *Unsupervised Learning* to form three clusters of records – Match, Non-Match and Potential Match. Then, those clusters will be the training data, since the records are now labeled. The results show that both the models tested surpass the probabilistic record linkage model.

3.3 Census Works

The BigMatch system was developed and is used by US Census Bureau [32]. It uses two files, **Record** file and **Memory** file, see Figure 3.2. The **Record** file is a very large file and the **Memory** file is medium size file that fits in the core memory. Then is applied various blocking criteria to the **Memory** file records. These blocking keys indexed. Following, while the method reads the **Record** file, it calculates each blocking key for every record and then searches in the index of the **Memory** file. If exists, a matching comparison is made between the fields. If that comparison is above a threshold the **Record** file record is saved in a subfile addressed to that blocking criteria. This subfile contains records that are plausible matches of records in **Memory** file. This way, the large **Record** file is read only once. Reportedly, BigMath could match 300.000 records per second, using a 10 blocking criterion.

United Kingdom Census also uses administrative data [33]. The methodology is described in Figure 3.3. First they do data cleaning and normalization and afterwards they generate various blocking keys (Matchkeys in their vocabulary). Next, they need to anonymize data, including the blocking keys, due to privacy concerns using cryptographic hash function, SHA-256 hash. Before the anonymization, they calculate the score of similarity between the fields. They use the SAS proprietary SPEDIS edit distance metric as the similarity metric. Now, they only have access to the encrypted fields and their similarity score, as well. Further, they match the records. First, through the blocking keys. If there is only one pair on the block, it is a match. If there is more pairs of records within the block then they use a logistic regression to decide. Before the logistic regression, they perform a selection of the candidates, based on some similarities and afterwards the logistic regression retrieves a probability. If the probability is equal or above 0.5 it is a match, otherwise it remains as unmatched.

Yancey compares some versions of the Jaro-Winkler, edit-distance metrics and even a Hybrid of



Records	Blocking Key
R1	Pereira1350358
R2	Rui1987

Pereira1350358		
First Name	Last Name	Postal Code
Manuel	Pereira	1350-358
Manel	Pereira	1350-358

Rui1994		
First Name	Last Name	Date of Birth
Rui	Silva	22-08-1994
Rui	Silves	22-05-1994

Figure 3.2: BigMatch Schema with the Record and Memory file

both [19]. The results show that the Hybrid metric was slightly better although it is a lot slower.

Another interesting work is happening in New Zealand Census with their study about administrative data based census [34]. The Census department of New Zealand is in the same phase of SP, studying the possibility of census based on administrative data. They have common methods with SP as, for instance, the validation phase where all data is standardized. The variables must be in the correct order and the same format as well as check if data contains duplicates and remove them.

Similarly, the personal identifiers are also encrypted although the variables are not anonymised which helps a lot in linking/matching records.

They use Standard Blocking to reduce the number of comparisons and use different blocking keys to ensure the errors in the attributes used to form a blocking key in order to not miss any links. For example, first they use as blocking criteria the attribute date of birth and link/match the data and after use different blocking criteria like the Soundex on the first and last name to try to link the rest of records that were impossible to link with the previous blocking criteria.

To decide if two records are the same they use the Fellegi-Sunter [5] method with two parameters – reliability (m) and commonality (u). The equations for agreement and disagreement weight are shown in Equation 3.2 and Equation 3.3.

$$\text{Agreement Weight}(m, u) = log_2 \frac{m}{u} \tag{3.2}$$

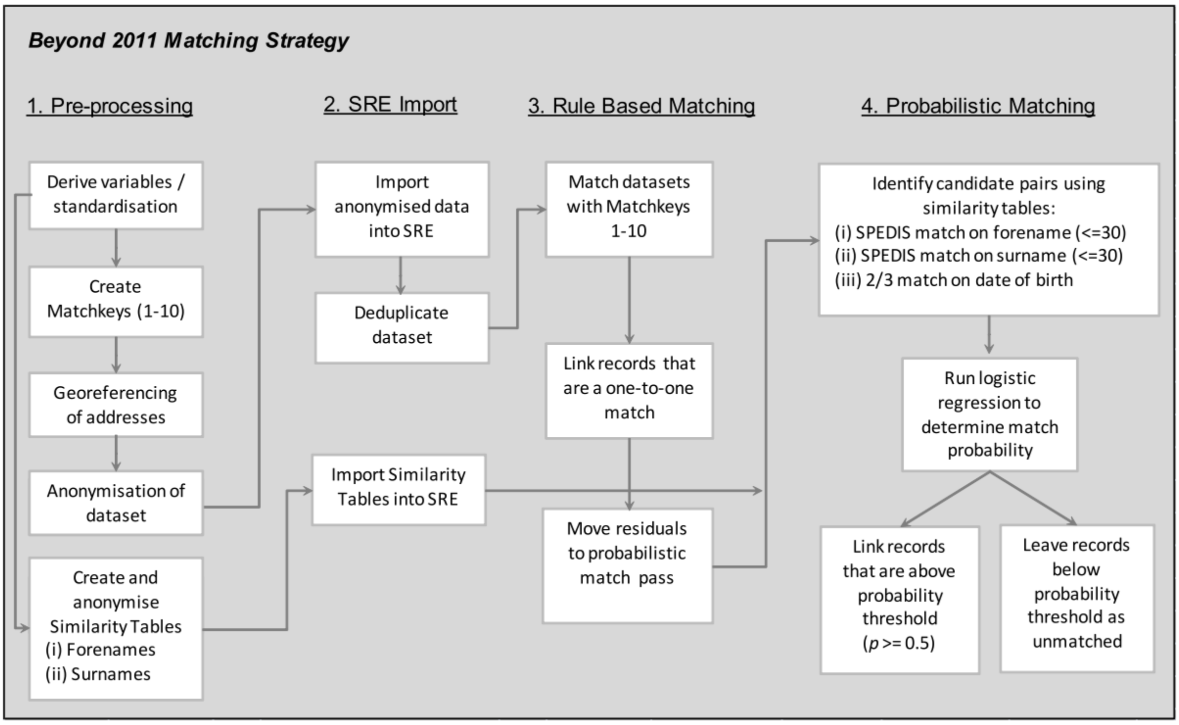

Beyond 2011 Matching Strategy
1. Pre-processing
2. SRE Import
3. Rule Based Matching
4. Probabilistic Matching
Derive variables / standardisation
Create Matchkeys (1-10)
Georeferencing of addresses
Anonymisation of dataset
Create and anonymise Similarity Tables
(i) Forenames
(ii) Surnames
Import anonymised data into SRE
Deduplicate dataset
Import Similarity Tables into SRE
Match datasets with Matchkeys 1-10
Link records that are a one-to-one match
Move residuals to probabilistic match pass
Identify candidate pairs using similarity tables:
(i) SPEDIS match on forename (<=30)
(ii) SPEDIS match on surname (<=30)
(iii) 2/3 match on date of birth
Run logistic regression to determine match probability
Link records that are above probability threshold (p >= 0.5)
Leave records below probability threshold as unmatched

Figure 3.3: Methodology for the UK Census

$$\text{Disagreement Weight}(m, u) = log_2 \frac{1-m}{1-u} \tag{3.3}$$

To calculate the weight m (reliability) they look at previous links and calculate the following probability:

$$Reliability = Prob(\text{two values agree} \mid \text{the records are a match}) \tag{3.4}$$

Additionally, to calculate the u (commonality) they look at the 100 000 most common values for each variable and use the following equation to calculate the probability:

$$Commonality = Prob(\text{two values agree} \mid \text{the records are not a match}) \tag{3.5}$$

In this context, to match two records each variable is compared through a metric (this metric is specific to each variable) and this metric chooses if the variable agrees or disagrees. If the variables agree it is used the Equation 3.2 with the pre-calculated weights - m and u. Otherwise, it is used the Equation 3.3. After all variables have the score, they are sum up to a final one. In the end, the final score has to be higher than a threshold to be considered a match.

3.4 Summary

It is important to understand these works in order to choose the best methods for our work. It is also good to know the methodology, especially in the Census works because it is very similar to this work and they already studied the problem and have a working solution with proven results.

4

Proposed Solution

Contents

The problem addressed in this work was approached as a classification problem. Since this was a two class problem (Match or Non-Match) we chose one of the simplest, but effective algorithms. The algorithm chosen was the *Logistic Regression* due to its simplicity. Also it is a method that runs fast and after some preliminary testing it presented positive results.

One of the main issues of this work is the number of comparisons to perform due to the millions of records to match. As shown in Section 2.1, Standard Blocking, reduces the number of comparisons by only comparing the records inside the same block. For this reason, we used this technique in order to be possible compare less records without compromising quality, by this means, true matches.

4.1 Solution Architecture

The proposed solution has two phases: first, the **Learning or Training Phase**, where the matching model is generated by training with some labeled data, followed by the **Testing or Classification Phase**, where unmatched records are classified as match or non-match. The architecture is represented in Figure 4.1. The process starts with the selection of pairs of databases to match, for instance, BDIC and IISS, see Table 4.1. Next, we need to perform some **Data Cleaning and Normalization** in order to keep only the same fields for both databases, with the same representation. Normalization is crucial because we need to select only the common fields and these fields must have the same nomenclatures/representations so the data is consistent. In regard to data cleaning, the data is cleaned in most fields, although the field – *Locality of Residence* – has many special characters (e.g. $\sim$) that we have removed.

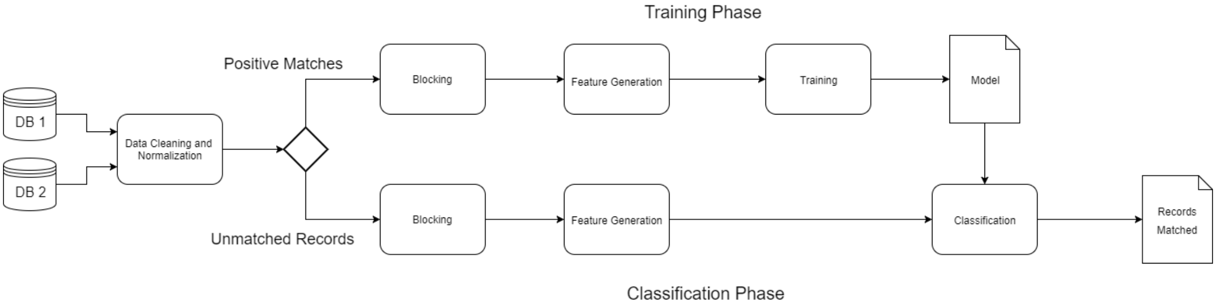


Figure 4.1: Record Linkage System Architecture

Following, we need to train a model capable of classifying pairs of unmatched records in matches or non-matches. Therefore, we use labeled data to train the model. The labeled data is obtained by joining the databases through the common key (or with an intermediate database, in the absence of one common key), to obtain the **Positive Matches**. In this context, I will continue with the example of BDIC and IISS. Then, we join BDIC and IISS through the common key, NIC. Now, we have two sets, the

Table 4.1: Records of BDIC and IISS

RECORDS	NIC	FIRST NAME	LAST NAME	DATE OF BIRTH
r_1	K100	RUI	LVA	30/05/1989
r_2	K101	RUI	UES	30/05/1989
r_3	K102	MAN	ZES	05/10/1995
r_4	K102	MIG	ECO	19/01/1999

RECORDS	NIC	NISS	FIRST NAME	LAST NAME	DATE OF BIRTH
s_1	K100	L564	RUI	LVA	30/05/1989
s_2	K101	L467	RUI	UEZ	30/05/1989
s_3	-	L321	MAN	SES	05/10/1995
s_4	-	L134	MIG	VAO	19/01/1999

Positive Matches (in the example the pairs r1, s1 and r2, s2) to train the model and the **Unmatched Records** that we want to match (records r3 and r4 with s3 or s4).

Taking the first set, it starts the **Training Phase**. So in this phase we apply the method Standard Blocking to create pairs of records between the two databases. With this step we keep the positive examples and generate negative examples. Thus, inside the same block, it has at most one true match and could have zero or more true non-matches. These true non-matches are perfect negative examples for the learned model. As a matter of fact, they are excellent because the model learns only from real examples, instead of generating negative examples and these real examples are more likely with the ones from the unmatched records because we use the same blocking key. Hence, the records within the same block could have the same errors and characteristics on the training set as the unmatched records have. The blocking criteria that we usually use is First Name + Date of Birth, see Table 4.2, where records r_1 through r_4 and s_1 through s_4 are the same shown in Table 4.1. Now, we have to form pairs between the records of BDIC and IISS, case they have the same blocking key. In this example, all records that match through the key NIC, have the same blocking key to show how negative examples are created. Thus, the records r_1 and r_2 go to the same block that s_1 and s_2. Pairs of records are represented as values of the similarities between these fields.

The similarity metric we use is the Edit Distance. In Table 4.3 are the scores of Edit Distance be-

Table 4.2: Blocking keys for all BDIC and IISS records

RECORDS	BLOCKING KEY
r_1	RUI30051989
r_2	RUI30051989
r_3	MAN05101995
r_4	MIG19011999

RECORDS	BLOCKING KEY
s_1	RUI30051989
s_2	RUI30051989
s_3	MAN05101995
s_4	MIG19011999

tween the pairs of records that were in the same block (**Feature Generation** step). Notice the negative examples generated with blocking on Table 4.3 like the pairs r_1, s_2 and r_2, s_1.

Table 4.3: Similarity scores of the fields of the matched records compared

RECORDS COMPARED	FIRST NAME	LAST NAME	DATE OF BIRTH	LABEL
(r_1, s_1)	0	0	0	MATCH
(r_1, s_2)	0	3	0	NON-MATCH
(r_2, s_1)	0	3	0	NON-MATCH
(r_2, s_2)	0	1	0	MATCH

Training the classifier is the last step of the first phase (**Training** step on Figure 4.1) resulting in a model capable of classifying unmatched records.

The **Classification Phase** starts by applying the same blocking method, Standard Blocking, with the same blocking criteria on the **Unmatched Records**. The **Unmatched Records** are the remaining records from each database that were not matched through the common key.

Following, we pair the records to be compared through the blocking and then we apply the Edit distance to each pair of fields like in the **Training Phase** (**Feature Generation** step), see Table 4.4.

Table 4.4: Similarity scores of the fields of the unmatched records compared

RECORDS COMPARED	FIRST NAME	LAST NAME	DATE OF BIRTH	LABEL
(r_3, s_3)	0	1	0	?
(r_4, s_4)	0	3	0	?

Now, the model can receive these values returned by the string similarity metric and classify each pair of record in match or non-match (**Classification** step), see Table 4.5.

Table 4.5: Classification of the records

RECORDS COMPARED	LABEL
(r_3, s_3)	MATCH
(r_4, s_4)	NON-MATCH

Finally, we have to apply some queries to the matches retrieved. First, we remove the non-matches results because we only want the matches and then we calculate how many of these matches, SP already matched. Second, we remove the matches previously matched by SP because we are interested only in new matches. By now, we only have new matches but since we used blocking there are matches of many to many. This means that the same record of BDIC could pair with different records of IISS and vice-versa. Thus, we only retrieve the pairs with the maximum probability of matching. For instance, if the same record of BDIC matches with two different records of IISS, we remove the pair with the lower probability of matching as it probably is a false positive. In the case that two or more pairs have the same higher probability, we keep all. We choose to keep both records in order to be able to do some clerical matching if necessary, although we know that only one is probably the right match.

4.2 Monitoring

In Chapter 2, one of the steps of record linkage was the evaluation of the process. This step is fundamental to understanding and confirming the results. This section presents the monitoring modules of the system.

4.2.1 Data Cleaning and Normalization

This module retrieves information about each database after the data cleaning and normalization. For instance, Table 4.6 shows some information about BDIC 2015, where the column completeness refers to the non-null attributes for each field in comparison with the total of records and is calculated according with the following Equation 4.1.

$$Completeness = \frac{\text{Number of Records Not Null}}{\text{Total of Records}} \tag{4.1}$$

The next column has the number of null values for each field and the last column the number of distinct values for each field.

The analysis of these values is essential to choose the right fields to perform the record linkage.

4.2.2 Quality of Blocking

We used Blocking to reduce the number of comparisons between two databases. With this module is possible to know how many of the records with blocking are positives (true matches) or negatives (true non-matches) in the case of train data for the model.

Another information that is useful is the Reduction Ratio of Blocking, i.e. the number of comparisons that we perform with blocking compared with the number of comparisons of a Cartesian product. Equation 4.2 shows the Reduction Ratio formula.

$$Reduction - Ratio = 1 - \frac{\text{Number of Comparisons}}{\text{Total of Records of Database 1} \times \text{Total of Records of Database 2}} \tag{4.2}$$

Table 4.6: Number of Records of BDIC 2015 and Completeness, Number of Null Records and Distant Values for each Field in BDIC 2015

Table 4.6 a) Number of BDIC Records

DATABASE	YEAR	NUMBER OF RECORDS
BDIC	2015	11 825 786

Table 4.6 a) Analysis on Completeness, Number of Null Records and Distant Values for each Field in BDIC 2015

FIELD	COMPLETENESS	NULL RECORDS	DISTINCT VALUES
PROV	100	0	1
ID	100	0	11 825 786
FIRST NAME	100	0	4251
LAST NAME	99.99	125	4543
SEX	100	0	2
BIRTH YEAR	100	0	148
BIRTH MONTH	99.99	54	14
BIRTH DAY	99.99	54	33
MARITAL STATUS	100	0	6
NATURALITY DISTRICT	100	0	33
NATURALITY COUNTY	100	0	361
NATURALITY PARISH	100	0	3712
NATIONALITY CODE	99.99	5	3
RESIDENCE DISTRICT	100	0	31
RESIDENCE COUNTY	100	0	310
RESIDENCE PARISH	100	0	3373
ZIP CODE 4	76.86	2 736 644	642
ZIP CODE 3	76.86	2 736 644	981
LOCALITY RESIDENCE	76.86	2 736 644	5363
RESIDENCE CODE	100	0	3

For example, to link two databases with 1000 records each, it would be necessary 1 000 000 comparisons between the records. By using blocking, the number of comparisons is 50 000. In this case, the reduction ratio would be $RR = 1 - \frac{50000}{1000000} = 0.95$. 95% of reduction.

4.2.3 Quality of the Models

As described in Section 2.4 there are some useful metrics that we can use to evaluate this work. To calculate the evaluation metrics to each model, we perform a 2-fold cross-validation over the records that matched through a common key. Applying Standard Blocking to these matches we get not only positive but also negative examples labeled. Therefore, half of that set is used to train a model, in order to classify the other half. The results of the matches are compared with the actual records labeled allowing to calculate the precision, recall and f-measure metrics.

For example, to calculate the precision score on the matches, we need the number of the matches classified by the model intersected with the labeled records and also the number of the records labeled as match. Thus, if the total of records labeled as match is, for instance, 100 000 and the records that the model classified as a true match are 80 000 – $Precision = \frac{80000}{100000} = 80\%$

4.2.4 Quality of Classification

In the previous Subsection, the metrics returned by each model show us how each model behaves with the train data but, is only a prediction about how the model will behave with the unmatched records. Hence, we need to have a control about the quality of the results, as well.

For this reason, the monitoring module retrieves information to analyze the matches. In Table 4.7, is the results of BDIC and AT where we have some metrics for each field. In first column, is number of contradictions between two records for a specific field. A contradiction happens when the values of the records are different. The next column is the percentage of contradictions for each field, calculated with the following Equation 4.3.

$$Contradiction = \frac{\text{Number of Records with different values}}{\text{Total of Records}} \tag{4.3}$$

The third column presents the number of uncertainties, that is, the number of records that have a null value. For instance, if a record has the Nationality field of BDIC is null, it is an uncertainty because we do not know if it is equal to the other. The same happens if it is the field of AT or both null. The Equation 4.4 describes the formula for uncertainty.

$$Uncertainty = \frac{\text{Number of Records with null values}}{\text{Total of Records}} \tag{4.4}$$

Table 4.7: Monitoring on BDIC 2015 and AT 2015 matches

Table 4.7 a) Number of Matches

MATCHES	YEAR	NUMBER OF RECORDS
BDIC AT	2015	244 903

Table 4.7 b) Analysis on the Contradictions and Uncertainties for each field on the Matches

FIELD	NUMBER OF CONTRADICTIONS	CONTRADICTION (%)	NUMBER OF UNCERTAINTIES	UNCERTAINTY (%)
FIRST NAME	0	0	0	0
LAST NAME	2 527	1.03	1	0.00041
SEX	1 380	0.56	0	0
BIRTH YEAR	0	0	0	0
BIRTH MONTH	0	0	0	0
BIRTH DAY	0	0	0	0
NATURALITY DISTRICT	14 210	5.80	0	0
NATURALITY COUNTY	14 475	5.91	0	0
NATURALITY PARISH	15 244	6.22	0	0
NATIONALITY CODE	2 664	1.09	0	0
RESIDENCE DISTRICT	244 903	100	0	0
RESIDENCE COUNTY	244 903	100	0	0
RESIDENCE PARISH	244 903	100	0	0
ZIP CODE 4	3 625	1.48	61 206	24.99
ZIP CODE 3	5 163	2.11	61 206	24.99
LOCALITY RESIDENCE	81 191	33.15	62 091	25.35
RESIDENCE CODE	7 050	2.88	0	0

We chose the blocking criteria – first name + date of birth – so these fields do not have any contradiction or uncertainty. For the rest of the fields is important to verify where exist more contradictions or uncertainties. The field ”*RESID_LOCAL_POSTAL*” has around 33% of contradictions due to different representations of the same place. For instance, the locality - *Santo António* - could be represented as – *St. António*. This difference is one of the advantages of the probabilistic method in regard to the exact methods. This contradiction in this field is not a particularity of this two databases but among all the databases.

4.3 Summary

This Chapter presents the solution architecture, as well as, the process of Record Linkage methodology for this work.

We implement a Monitoring module for different parts of the methodology. Monitoring is very important to the whole process since the beginning, to control the quality and have a better comprehension of the results.

5

Results

Contents

In this Chapter is presented the results achieved in this project.

5.1 Experimental Setting

In this Section we describe the tools and data used for this work.

The Databases are stored in an Oracle Server and we perform queries through the graphical tool SQL Developer. We access SQL Developer through a Linux server (Debian) with 32GB of RAM and 2 CPU's Intel(R) Xeon(R) CPU X5680 3.33GHz.

The programming language used for training a model and classifying the records was Python. In the code we used the logistic regression from the scikit learn library with a l2 penalty for the model and without sample weights and the Edit Distance from the levenshtein library. To load the records from a CSV file (downloaded from SQL Developer) we used the Pandas Dataframe. We also used BitBucket for version control of the code.

Moreover, to train each model we used the respective databases that we want to match. However, in the case of SEF and EDUC the number of matched records was only 7 885 (see Table 5.3). As a consequence of the low number of matches, we opted to generate a new model with SEF and IISS the same fields that EDUC because it has more records to be trained. Of course it would be better to use a model between SEF and EDUC, but using other did not compromised the quality of the results.

Further, to train a model for the databases BDIC and AT we need positive examples. The problem is that is not possible to find matches between BDIC and AT because these databases do not share a common personal identifier/key. For this reason, we used the database IISS to be an intermediate between these two databases, since IISS has the two keys - NIC and NIF.

In Table 5.1 are the pairs of records that Standard Blocking generate in comparison with all the comparisons necessary if did not use an indexing technique. Finally, in last column is the Reduction Ratio calculated.

It is incredible to have a reduction over 99% in all databases, even when Standard Blocking generates millions of records. The results on Table 5.1 demonstrate the reason why we need to use Standard Blocking.

The blocking criteria used for the majority of databases was *First Name* + *Date of Birth* due to the good results achieved on the quality of the models. However, to match SEF with IISS we noticed that if we used a different blocking key, we could find more matches per block. For this reason and only for SEF and IISS we used the blocking key *Birth Country* + *Date of Birth*.

Table 5.1: Comparison between the number comparisons with and without Standard Blocking

DATA SOURCES	NUMBER OF RECORDS	RECORDS WITHOUT COMMON KEY	NUMBER OF COMPARISONS WITHOUT BLOCKING	NUMBER OF COMPARISONS WITH BLOCKING	REDUCTION RATIO (%)
BDIC 2015	11 825 786	6 933 267	3.09×10^{13}	61 038 705	99.9998
AT 2015	9 370 879	4 414 595			
BDIC 2015	11 825 786	6 283 141	8.7×10^{12}	44 959 327	99.9995
IISS 2015	6 927 720	1 385 062			
BDIC 2015	11 825 786	10 230 736	8.69×10^{11}	720 233	99.9999
EDUC 2015	1 680 018	84 968			
BDIC 2015	11 825 786	11 203 211	7.12×10^{11}	2 508 243	99.9996
IEFP 2015	686 198	63 622			
BDIC 2015	11 825 786	11 014 943	2.31×10^{12}	5 118 284	99.9997
CGA 2015	1 032 133	209 642			
SEF2015	383 764	253 742	1.58×10^{11}	191 103	99.9998
IISS 2015	6 927 720	624 118			
SEF 2015	383 764	220 315	1.99×10^{12}	1 297 139	99.9999
AT 2015	9 186 325	9 023 088			
SEF 2015	383 764	375 872	2.97×10^{10}	37 238	99.9999
EDUC 2015	87 017	79 132			

5.2 Results for the Quality of the Models

Before we use the models to classify records, we check the quality through some metrics. In Table 5.2 is presented the precision, recall and f-measure for both Matches and Non-Matches.

In general, the results are good. The scores for each metric have, usually, high values. However, there are some exceptions like the model BDIC 2015 and EDUC 2015. The reason to have a lower score, compared with the rest, is mainly because of the quality of the data of EDUC 2015. There are a lot of errors in a lot of fields that we checked in matches between BDIC 2015 and EDUC 2015. Moreover, there are a lot of null values on the fields in common with BDIC 2015, the ones that are used to compare and match. Additionally, in the model BDIC 2015 and CGA the scores for the precision and consequently f-measure, are lower for the same reasons that BDIC 2015 and EDUC 2015.

5.3 Matching Results

Table 5.3 presents the number of records that we start working in the beginning. Our job was to match each pair of the databases presented in Table 5.3, where the second column shows the number of records of the respective database and the third column shows the key used to match records. The next column presents the number of match records through the key. These match records are the ones that will be used to train a model. Finally, in the last column are the unmatched records. Our goal is to find a connection between each pair of unmatched records.

One of the requirements to perform this work was to know and understand each database and the relations between the others. For instance, the database BDIC contains only Portuguese citizens,

Table 5.2: Metrics of evaluation for each model

DATA SOURCES MATCHED	MATCHES			NON-MATCHES		
	PRECISION (%)	RECALL (%)	F-MEASURE (%)	PRECISION (%)	RECALL (%)	F-MEASURE (%)
BDIC 2015	98	97	97	97	98	97
IRS 2014						
BDIC 2015	99	98	99	100	100	100
IISS 2015						
BDIC 2015	93	94	93	99	99	99
EDUC 2015						
BDIC 2015	99	98	98	99	99	99
IEFP 2015						
BDIC 2015	88	95	91	99	99	98
CGA 2015						
SEF2015	98	97	98	98	98	98
IISS 2015						
SEF 2015	97	97	97	97	98	97
AT 2015						
SEF 2015	95	96	96	97	96	97
EDUC 2015						

Table 5.3: Initial number of records for each pair of databases

DATA SOURCES MATCHED	NUM. RECORDS	KEY USED	RECORDS MATCHED THROUGH COMMON KEY	RECORDS WITHOUT COMMON KEY
BDIC 2015	11 825 786	NIC	4 892 526	6 933 267
AT 2015	9 370 879	NIF		4 414 595
BDIC 2015	11 825 786	NIC	5 542 658	6 283 141
IISS 2015	6 927 720	NIC		1 385 062
BDIC 2015	11 825 786	NIC	1 595 050	10 230 736
EDUC 2015	1 680 018	NIC		84 968
BDIC 2015	11 825 786	NIC	622 576	11 203 211
IEFP 2015	686 198	NIC		63 622
BDIC 2015	11 825 786	NIC	810 843	11 014 943
CGA 2015	1 032 133	NIC		209 642
SEF2015	383 764	NISS and NIF	118 155	253 742
IISS 2015	6 927 720	NISS and NIF		624 118
SEF 2015	383 764	NIF	163 237	220 315
AT 2015	9 186 325	NIF		9 023 088
SEF 2015	383 764	NISS and AR	7885	375 872
EDUC 2015	87 017	NISS and AR		79 132

Table 5.4: Number of Records added by our probabilistic method

DATA SOURCES MATCHED	RECORDS TO MATCH	NEW MATCHES
BDIC 2015	6 933 267	244 903
AT 2015	4 414 595	
BDIC 2015	6 283 141	47 836
IISS 2015	1 385 062	
BDIC 2015	10 230 736	51 138
EDUC 2015	84 968	
BDIC 2015	11 203 211	11 974
IEFP 2015	63 622	
BDIC 2015	11 203 211	60 545
CGA 2015	209 642	
SEF2015	253 742	30 120
IISS 2015	624 118	
SEF 2015	220 315	52 177
AT 2015	9 023 088	
SEF 2015	375 872	12 796
EDUC 2015	79 132	

thereby it would be wrong to match Portuguese people with foreign people of the SEF database. We apply the same principle when we do record linkage between SEF and the others databases by only choosing the foreign people. We used two keys when matching SEF, IISS and SEF, EDUC because it is possible to find more positive examples although on SEF and EDUC the number of positive examples was too low and we had to use a different model.

After generating a model of the pair of databases that we are trying to match is time to use it for the classification phase. In Table 5.4, the results are presented . Basically, we classified each pair of unmatched records as match or non-match.

Our results are within expectations. We could find thousands of new matches which was our ultimate goal. We did not find all the links between the records because it is not a trivial task but in Section 6.1 is some improvements and ideas to have better results.

SP already linked most of the records and some databases are not supposed to link with BDIC or SEF totally. For example, the database EDUC contains also foreign people, thus, those 84 968 records to match are not supposed to match totally with BDIC.

The first linking was between BDIC and AT and the result was very good. The reason to the high number of matches retrieved is because the data in each database is very clean, with few errors and null attributes.

These results are not final and SP will decide if they are, in fact, a match and if so, will apply the residence rules to decide if the person lives in Portugal or not. If the person lives in Portugal, the linked record is stored into BPR.

Another interesting way of looking at the results is to calculate the additional matches over the records

Table 5.5: Analysis on the new matches in comparison with the not integrated records in BPR

DATA SOURCES MATCHED	RECORDS NOT INTEGRATED IN BPR	ADDITIONAL MATCHES (%)
BDIC 2015	401 829	244 903 (60.95)
AT 2015		
BDIC 2015	248 953	47 836 (19.21)
IISS 2015		
BDIC 2015	110 480	51 138 (46.29)
EDUC 2015		
BDIC 2015	30 268	60 545 (200)
CGA 2015		
SEF2015	248 953	30 120 (12.1)
IISS 2015		
SEF 2015	401 829	52 177 (13)
AT 2015		
SEF 2015	110 480	12 796 (11.6)
EDUC 2015		

that were not possible to integrate into BPR, by SP. In Table 5.5, is represented the number records that SP could not match for each database and, thereby, are not in the BPR and also the number of our additional matches with the respective additional percentage. This Table completes Table 5.4 showing the additional records matched with BDIC and SEF with each database.

In the case of IEFP, we linked 11 974 records from the 63 622 unmatched records. The reason why is not in 5.5 is because SP had a problem of duplicates on the database of IEFP 2015 and that lead to more matches that were necessary and for that reason is not possible to compare.

Our links between BDIC and CGA are superior to the records not integrated in the BPR because we have some duplicates but also because we paired with more records than we should. A possible cause for this error is the precision for the matches. This value was the lowest of all models (88%). One solution to this problem is to create a threshold to remove the extra matches. Nevertheless, SP have the final word about the new matches and how to choose them.

The results are good, in general. Notice that not all the records that are not integrated into BPR should pair. The BPR records are only for residents in Portugal. For this reason, if a Portuguese citizen (that is in the BDIC database) is living abroad, obviously, he or she will not appear in EDUC, IEFP or CGA, for example. The same happens for the foreign people in SEF.

5.4 Comparison with Statistical Portugal

One way to evaluate our work is to compare the results obtained with the Machine Learning approach used against the exact matching methods previously used by SP. When we perform record linkage

Table 5.6: Comparison with SP results

DATA SOURCES MATCHED	RECORDS MATCHED THROUGH COMMON KEY	OUR MATCHES IN BPR	RECORDS MATHED BY SP	RECORDS VALIDATED (%)
BDIC 2015 AT 2015	4 892 526	3 262 651	3 843 574	84,89
BDIC 2015 IISS 2015	5 542 658	582 237	851 212	68,4
BDIC 2015 EDUC 2015	1 595 050	8 224	40 769	20,17
BDIC 2015 IEFP 2015	622 576	55 260	89 587	61,68
BDIC 2015 CGA 2015	810 843	168 158	180 453	93,19
SEF2015 IISS 2015	118 155	2 249	25 795	8,72
SEF 2015 AT 2015	163 237	7 990	22 368	35,72
SEF 2015 EDUC 2015	7 885	2 691	8 963	30,02

between the unmatched records of two databases, we often retrieve matches already matched by SP. Table 5.6 shows the number of matches through the common key, the intersection of our matches and the SP matches and in the fourth column the matches of SP, without the matches with the common key. The last column has the percentage of our matches compared with SP matches.

The results for some pairs of databases are really good but for others not so much. This happens because the record linkage was performed by pairing only two databases at a time and without using information of previous matches as SP did. For example, doing record linkage between BDIC and IISS will find new matches and consequently new keys and attributes. The new keys could find new matches easily when linking with other databases and the new attributes help a lot on the record linkage.

5.5 Expert Evaluation

While performing this work, SP asked an expert to validate the results in order to check the data. The expert had access to the new and more updated version of IISS, namely the version of 2016 (we used the version of 2015 because was the most recent at the time). BDIC and AT do not share a common key but IISS has both keys. Thus, is possible to join BDIC and AT using as intermediate IISS, like we did with when we paired the databases of 2015. The version of IISS 2016 is better and has new connections between NIC (key of BDIC) and NIF (key of AT) that were unavailable in the version of 2015, allowing to check if our matches are correct.

The first analysis was to verify if we have different genders in our matches. For example, if a record from BDIC is male, check the correspondent gender of AT is also male. The total of matches is 246 217 which is a higher number of records to the matches presented in Table 5.4 because we delivered to the expert all matches and not only the ones with maximum probability to not exclude any potential match.

The results of the gender analysis show that in the 246 217 matches, 244 721 have the same sex in BDIC and in AT which correspond to 99,39%. This is a good result because the sex of a person is a good identifier and except for errors in the data or changes of sex, that is very improbable, the sex shows us that the pairings with our method are consistent.

Nevertheless, this test is not enough to confirm the matches. Thus, the expert used the IISS 2016 to compare. He joined our matches through NIC and NIF and discovered 10 454 records that have the same NIC and NIF linked as we did from the 10 497 records he could join. That is 99.59% of correct matches. Although is just a small portion of the 246 217, it proves that those matches are correct and also that the rest of the matches that were not found, could still be unmatched in the IISS 2016.

5.6 Summary

This Chapter presents the results achieved in this work. Before the results there is some context in the environment that we work at SP.

The Reduction Ratio shows the importance of using Standard Blocking. Even using Standard Blocking we have to do millions of comparisons but, compared with a Cartesian product, it is less than 1% of comparisons.

Before pairing two databases it is important to calculate the Quality of the Models to check if it has the necessary qualifications to match the databases. The metrics retrieved have high values for most of the models.

The additional matches we pair show that our methodology worked as expected. In the case of BDIC and CGA we retrieved more matches than expected and the cause could be the quality of the model.

The intersection of all our matches with SP matches were always below 100%. Some pairings we have higher values and in others low. This happens because SP matched with other databases and was able to retrieve more information like keys and new attributes.

Finally, the evaluation by an expert prove that some of our matches are correct and the rest could be also a true match.

6

Conclusion

Contents

In this chapter, we provide an overview of this work and a reflection of what was positive and negative. Following, in Section 6.1 is ideas and thoughts of what could be improved and new methods of performing this work.

One of the most important things this project taught was that Data Cleaning and Normalization is crucial in Record Linkage. Because is the first step, it has to be performed carefully, otherwise, it would lead to major implications in the following steps of the methodology. With the knowledge of today, it would be a step that I would pay more attention. However, while performing this work, we had a tight schedule.

In terms of our methodology, although it is still not perfect, it is solid and the results are a proof of that. In less than a year we found thousands of new links/matches for different pairs of databases. These results will help SP have a better BPR, although a lot of records are still unmatched. When we first started, we have the notion that we would not find all the matches because of the data itself. Some databases have a lot of null values that hinder the matching, not to mention the anonymization on the first and last name that is the principal difficulty of linking records.

Nevertheless, every year, each database will, hopefully, have more accurate and cleaner data and, therefore, will be easier to link the records. One example of this is the database of IISS 2016 that have more connections between two personal identifiers/keys.

The positive aspects of this work are that we have a solid methodology for the record linkage through probabilistic methods. The results prove that. Especially between BDIC and AT where we pair 244 903 new matches. Moreover, we have a Monitoring component that retrieves useful information to evaluate the data and results.

It was also important, the results from the evaluation from the expert, to prove that we found new matches.

From a high perspective it is a simple process but when we start to record linkage two different datasets there is always a new problem that arises. For example, some datasets do not have a key fulfilled to all records. This is a problem to identify the record when we check if a record is only matched with the other. Another problem is the fields to compare. Between different datasets, the common fields are different and sometimes have different representations. Furthermore, it was needed to have special attention to the ratio of positive and negative examples on the training set. In some datasets, after applying the blocking to the matched records we noticed that the number of matches was far superior to the non-matches. Ideally, this is the expected but, we have to increase the number of non-matches, doing a selection of the number of matches and non-matches, so the model generated is not biased and do not return an increased number of false negatives.

Another thing that hindered was the space for the databases that we could use. Since each database has millions of records and we need to create several intermediate databases to get the final result, it was difficult to manage the space since some databases contained important data. On the other hand, it let us have a more organized environment.

Also, the extra matches in CGA were not expectable. SP will decide what to do with the results and we hope that some matches could be integrated into the BPR.

To sum, the goals for this work were accomplished, we matched the databases and achieved good results, in general.

6.1 Future Work

This Section presents some thoughts about what could be enhanced in the present system.

To compare the records is essential to use an indexing technique and we used Standard Blocking. By using this technique, with a specific blocking key, we exclude potential matches. We always used the blocking criteria based on the first name + date of birth (except for the pairing of SEF and IISS). If a record from a database has an error on these fields, the record will never link with the other, from another database. Thus, one way of solving this problem is by using a different blocking keys with different blocking criteria to discover new matches.

We opted for using always the Edit Distance, although there are other similarity metrics. For example, for dates of birth, it could be used a different metric, as well as for the postal codes and for the other fields. Each metric should take into account the fields particularities. Another important factor is the value a similarity metric should return in the case of a field or both fields are null.

It is crucial to analyze every field of every record in search of anomalies. In the field Locality of Residence, we notice many special characters. The other fields could have some irregularities as well. Also, the Normalization should be improved to check if every field is represented equally. For instance, for the Marital Status field in SEF we noticed that the code for every status was different from the others databases. Also, if the zip code has the value 000 it means that it is null and the same happens for the Nationality country code for the value ZZ. These aspects are difficult to clean and normalize unless we have a clear understanding of the data and is very important to correct them to have accurate data to lead to accurate results.

The current process pairs databases to find new matches. These new matches add new fields and new keys to the current matches. The new information acquired can be used to find more matches in two ways: with the new keys is easy to join databases that also have the same key and find more correspondences and second, with the new fields, it is possible to use more fields to compare and in many cases complete the non-null fields .

By using the new matches to find more matches we can define an iterative method for finding even more matches.

Our method retrieves thousands of new matches that SP will decide if they are, in fact, matches. Because our method has a matching probability, we could calculate an optimal threshold for separating the true positives from the false/potential matches.

Last but not least, is to implement additional software to improve automation to the process. The more automated is the process, the faster we have the results. Another advantage is to minimize errors. We already have some scripts for the Monitoring process and also for the record linkage process. Although it is very complicated to have everything automated because every database has its peculiarities, it is possible to have more than we already have.

Bibliography

[1] "Metodologia de atualização da Base de População Residente - Construção da BPR 2015 (Work Document)," Instituto Nacional de Estatística, Tech. Rep., 2016.

[2] R. Silva, L. S. Velho, P. Calado, and M. J. Silva, "Matching Records for Census Data," *Data Science Statistics & Visualisation*, 2017.

[3] L. Velho, "Emparelhamento de dados Censitários," Master's thesis, Instituto Superior Técnico, 2017.

[4] D. A. Freedman, *Statistical Models: Theory and Practice*. Cambridge University Press, 2005.

[5] I. P. Fellegi and A. B. Sunter, "A Theory for Record Linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, p. 1183, 12 1969. [Online]. Available: http://www.jstor.org/stable/2286061?origin=crossref

[6] P. Christen, "A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 9, pp. 1537–1555, 9 2012. [Online]. Available: http://ieeexplore.ieee.org/document/5887335/

[7] M. Hernandéz and S. Stolfo, "The Merge / Purge Problem for Large Databases," *SIGMOD '95 Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pp. 127–138, 1995.

[8] A. Doan, A. Halevy, and Z. Ives, *Principles of data integration*. Morgan Kaufman, 9 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2347696.2347721

[9] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998. [Online]. Available: http://www.springerlink.com/index/Q87856173126771Q.pdf

[10] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[11] R. Board and L. Pitt, “Semi-supervised learning,” Ph.D. dissertation, Carnegie Mellon University, 10 1989. [Online]. Available: http://pages.cs.wisc.edu/~jerryzhu/pub/thesis.pdfhttp://link.springer.com/10.1007/BF00114803

[12] D. Yarowsky, “Unsupervised word sense disambiguation rivaling supervised methods,” in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics* -, 1995, pp. 189–196. [Online]. Available: http://portal.acm.org/citation.cfm?doid=981658.981684

[13] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” *Proceedings of the eleventh annual conference on Computational learning theory - COLT’ 98*, pp. 92–100, 1998. [Online]. Available: http://portal.acm.org/citation.cfm?doid=279943.279962

[14] B. Settles, “Active Learning Literature Survey,” *Machine Learning*, vol. 15, no. 2, pp. 201–221, 2010.

[15] S. Tong and E. Chang, “Support vector machine active learning for image retrieval,” *Proceedings of the ninth ACM international conference on Multimedia - MULTIMEDIA ’01*, no. C, p. 107, 2001. [Online]. Available: http://portal.acm.org/citation.cfm?doid=500141.500159

[16] J. Macqueen, “Some methods for classification and analysis of multivariate observations,” *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 233, pp. 281–297, 1967.

[17] T. Kohonen, “Self-organized formation of topologically correct feature maps,” *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[18] L. Vinet and A. Zhedanov, “Binary codes capable of correcting deletions, insertions, and reversals,” *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966. [Online]. Available: http://arxiv.org/abs/1011.1669http://dx.doi.org/10.1088/1751-8113/44/8/085201

[19] William E. Yancey, “Evaluating string comparator performance for record linkage,” *Statistical Research Division*, pp. 3905–3912, 2005. [Online]. Available: http://www.amstat.org/sections/srms/Proceedings/y2006/Files/JSM2006-000855.pdf

[20] S. Yan, D. Lee, M.-Y. Kan, and L. C. Giles, “Adaptive Sorted Neighborhood Methods for Efficient Record Linkage,” *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, p. 185–194, 2007. [Online]. Available: http://doi.acm.org/10.1145/1255175.1255213

[21] A. McCallum, K. Nigam, and L. L. Ungar, “Efficient clustering of high-dimensional data sets with application to reference matching,” *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 169–178, 2000. [Online]. Available: http://portal.acm.org/citation.cfm?id=347123

[22] A. E. Monge and C. P. Elkan, “An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records,” *Proceedings of the SIGMOD 1997 workshop on research issues on data mining and knowledge discovery*, pp. 23–29, 1997.

[23] R. Baxter, P. Christen, and T. Churches, “A Comparison of Fast Blocking Methods for Record Linkage,” *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Workshop*, pp. 25–27, 2003.

[24] M. Cochinwala, V. Kurien, G. Lalk, and D. Shasha, “Efficient data reconciliation,” *Information Sciences*, vol. 137, no. 1-4, pp. 1–15, 9 2001. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0020025500000700

[25] Liang Jin, Chen Li, and S. Mehrotra, “Efficient record linkage in large data sets,” in *Eighth International Conference on Database Systems for Advanced Applications, 2003. (DASFAA 2003). Proceedings.* IEEE, 2003, pp. 137–146. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1192377http://ieeexplore.ieee.org/document/1192377/

[26] C. Faloutsos and K.-I. Lin, “FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets,” *Proceedings of the 1995 ACM SIGMOD international conference on Management of data - SIGMOD ’95*, vol. 24, no. 2, pp. 163–174, 1995.

[27] S. Sarawagi, L. Breiman, J. H. Friedman, A. Richard, and C. J. S. Classification, “Interactive Deduplication using Active Learning,” *KDD ’02 Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining Pages 269-278*, 2002.

[28] P. Christen, *Data Matching.* Springer Berlin Heidelberg, 2012, no. Chapter 1. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-10876-5_5http://link.springer.com/10.1007/978-3-642-31164-2

[29] J. Wang and S. Madnick, “The inter-database instance identification problem in integrating autonomous systems,” in *Proceedings. Fifth International Conference on Data Engineering*. IEEE Comput. Soc. Press, 1989, pp. 46–55. [Online]. Available: http://ieeexplore.ieee.org/document/47199/

[30] H. Galhardas, D. Florescu, D. Shasha, E. Simon, H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C. S. Declarative, “Declarative Data Cleaning : Language , Model , and Algorithms,” INRIA, Tech. Rep., 2006.

[31] M. G. Elfeky, V. S. Verykios, and A. K. Elmagarmid, “TAILOR: a record linkage toolbox,” *Proceedings 18th International Conference on Data Engineering*, pp. 17–28, 2002.

[32] W. E. Yancey, “Big Match: A program for extracting probably matches from a large file for record linkage,” *Statistical Research Division*, vol. 1, no. 1, pp. 1–8, 2002. [Online]. Available: http://www.census.gov.edgekey-staging.net/srd/papers/pdf/rrc2002-01.pdf

[33] “Beyond 2011: Matching Anonymous Data,” Office for National Statistics, Tech. Rep. July, 2013.

[34] “Experimental population estimates from linked administrative data : methods and results,” Statistics New Zealand, Tech. Rep., 2016. [Online]. Available: http://www.stats.govt.nz/~/media/Statistics/surveys-and-methods/methods/research-papers/topss/exp-pop-estimates-linked-admin-data-methods-research/exp-popln-estimates-from-linked-admin-data.pdf