Matching Census Data Records

RUI SILVA, Instituto Superior Técnico, Portugal

Record Linkage is the task of matching two records that refer to the same entity. In Portugal, Statistics Portugal started a study to use administrative data in the Census. However, due to inconsistent and anonymised data, Statistics Portugal was unable to pair all the records. In this context, this work aims to match records of administrative databases for improving the process of the Portuguese data Census. This dissertation presents methods for record linkage taking into account effectiveness, efficiency and related Census works. Moreover, presents a record linkage system based on Supervised Learning as well as methods to evaluate the results. Our methodology led to an increase of the records matched where the best result was between BDIC (Civil Population Register) and AT (Tax Authority) by pairing 244 903 records which represent a 60.95% increase.

Additional Key Words and Phrases: Record Linkage, Census, Machine Learning, Logistic Regression

ACM Reference Format:
Rui Silva. 2017. Matching Census Data Records. 1, 1 (November 2017), 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Every ten years in Portugal, and in many other countries, Census are performed. Census is the biggest statistical operation 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 Census data are progressively obtained from Public Administration databases.

The Administrative Model carries some advantages, like reducing the costs, reducing 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 improve the actual model.

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) [INE 2016]. Currently, SP has access to more that 10 administrative databases. The administrative databases used in this work are:

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

The goal of SP is to match the records that refer to the same person between the databases. This will allow SP 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.

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 is 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.

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 this number.

Following, is the problem of keys/personal identifiers. Despite of the absence of a single key, we have some common keys among the databases. Nevertheless, many records may not have the common key totally filled. In this case, we need 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. Even when both keys are filled for every record, some problems may arise like: 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 dies and only one database is updated or a nationality change is registered only on one database. Finally, there is the hypothesis of errors on the records, even in the key field.

Author's address: Rui Silva, Instituto Superior Técnico, Av. Rovisco Pais 1, Lisboa, 1049-001, Portugal, ruimenezessilva@tecnico.ulisboa.pt.

Furthermore, the anonymisation and pseudonymisation imposed by the CNPD rises 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.

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. For instance, the same person in two different databases can have a typographical error in the last name, or a different name for the same city (like Porto and Oporto) or even the day and month switched. These errors and inconsistencies hinder the record linkage and are one of the main causes for SP not to find all true matches across the databases with exact methods.

1.2 Goals

The goal of this work was to design a record matching model, that will receive records from different databases and determine if the records refer to the same person. This will make possible for 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.

Our record matching model will not be through exact methods but instead through probabilistic methods. This matching model will have a component of monitoring as well.

SP already started the creation of the BPR. As a consequence of the problems referred before, they were unable to match them all. 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?

2 BACKGROUND

Record Linkage is the task of finding records in different databases that refer to the same entity, even if the records are not identical [Fellegi and Sunter 1969]. 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"* [Christen 2012a].

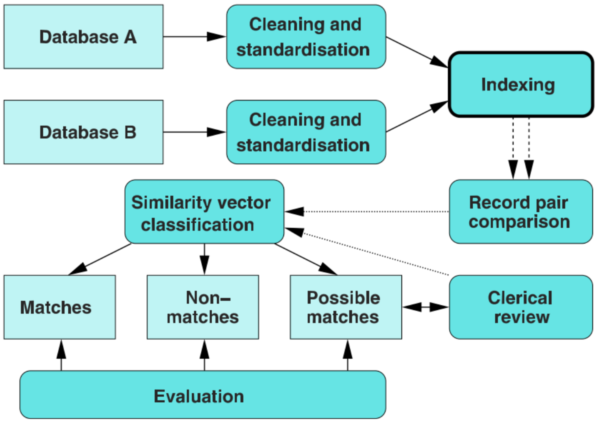

Fig. 1. Schema of record linkage process, taken from [Christen 2012a]

In Fig. 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.

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 **Stantdard Blocking** [Fellegi and Sunter 1969]. **Standard Blocking** consists of grouping records that are similar by using a blocking key. A blocking key is formed with the concatenation of one or more attributes. The blocking criteria defines the rule about how attributes are concatenated. Thus, to link two databases, for instance, after each record has a blocking key, blocks are formed by grouping all the records with the same blocking key. Therefore, records from one database that are in one block, will be compared only with the records from the other database within the same block. Inevitably, less comparison will be done with this method.

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 different types of machine learning models like **Supervised Learning** or **Unsupervised Learning**.

In **Supervised Learning** a training set is given, in which the data is labeled [Doan et al. 2012]. 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 unlabeled data and mark as match or non-match. Possible algorithms are Support Vector Machines [Burges 1998], Decision Trees [Quinlan 1986] and Logistic Regression [Freedman 2005], among others.

In contrast, **Unsupervised Learning** uses only unlabeled data [Doan et al. 2012]. 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 [Macqueen 1967] and Self Organized Maps [Kohonen 1982].

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 1) to calculate the probability of a given attribute or set of attributes corresponds to a defined class, normally set as 0 or 1.

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

The input values of the logistic function, are combined linearly using weights, normally represented as β. These weights or coefficients can be calculated with maximum-likelihood estimation using training data.

2.3 String Similarity Metrics

This Section presents some similarity metrics regularly used to compare strings. 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 is a string similarity metric that returns 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 [Vinet and Zhedanov 1966]. The formula is represented in Equation 2.

$$d(i,j) = min \begin{cases} d(i-1, j-1) + c(x_i, y_j) & \text{copy or substitute} \\ d(i-1, j) & \text{delete } x_i \\ d(i, j-1) & \text{insert } y_j \end{cases} \tag{2}$$

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

In Table 1 is an example of 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.

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

Table 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**

Jaro metric was developed primarily to compare first and last names [William E. Yancey 2005]. The formula is presented next:

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

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.

3 RELATED WORK

This section, presents some works focused on accomplishing efficiency or effectiveness in Record Linkage. As we saw in Fig.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. We now describe a few solutions that were proposed for this problem.

Yan et al. show two different approaches of the Sorted Neighbourhood Method (SNM) [Yan et al. 2007]. 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.

The comparison with Standard Blocking and SNM reveal that both IA-SNM and AA-SNM have better results.

McCallum proposes the use of canopies. Canopies are similar to clusters, the difference being that they are created with a cheap similarity measure and overlap each other [McCallum et al. 2000].

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 [Monge and Elkan 1997]. The structure used was an undirected graph in which the nodes are records and the edges between the nodes represent if they are duplicates.

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 [Cochinwala et al. 2001]. So there is a trade-off between complexity and classification accuracy.

A different approach is proposed by Jin et al. [Liang Jin et al. 2003], where the blocking key values are converted to a multidimensional Euclidean space, using a function named StringMap, a modification of FastMap [Faloutsos and Lin 1995]. 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

In this Section, some relevant techniques in the process of record linkage are presented .

The problem of comparing dates is addressed in [Christen 2012a], where these are compared based on the difference of days using numeric absolute difference presented in Eq. 4. 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.

$$sim_{day\_abs}(d_1, d_2) = \begin{cases} 1.0 - (\frac{|d_1 - d_2|}{d_{max}}) & \text{if } |d_1 - d_2| < d_{max} \\ 0.0 & \text{else} \end{cases} \tag{4}$$

Sarawagi et al. presented ALIAS [Sarawagi et al. 2002], 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 [Christen 2012b]. 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 [Wang and Madnick 1989]. 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, although requiring much tuning, a high knowledge of the data set by the expert and being a very complex task. As a consequence, machine learning is commonly used to create a model and afterwards, the generated rules are tuned.

Elfeki et al. present a record linkage toolbox - TAILOR [Elfeky et al. 2002] - 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

United Kingdom Census also uses administrative data [Sur 2013]. 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. 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. Following, 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 record 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.

[William E. Yancey 2005] compares some versions of the Jaro-Winkler, edit-distance metrics and even a Hybrid of both. The results show that the Hybrid metric was slightly better although it is a lot slower.

4 SOLUTION ARCHITECTURE

Our solution has two phases: first, the **Learning or Training Phase**, where the matching model is generated by training with some labeled data and second, the **Testing or Classification Phase**, where unmatched records are classified as match or non-match. The architecture is represented on Figure 2.

The process starts with selecting the databases to match. 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.

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. Primarily, we join the databases through the common key (or with an intermediate database, in the absence of one common key), to obtain the **Positive Matches**. Now, we have two sets, the **Positive Matches** to train the model and the **Unmatched Records** that we want to match.

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 model learn because they have the same characteristics that the non-matches from the unmatched records. The blocking criteria that we usually use is First Name + Date of Birth. Thus, after generate the blocking keys we form pairs between the records from the two databases that have the same blocking key.

The similarity metric we use is the Edit Distance. As explained in Section 2.3 the lower the value returned by the Edit Distance, more similar are the strings. So we apply this metric between each common field of the pair of records

Training the classifier is the last step of the first phase (**Training** step on Figure 2) 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).

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).

Finally, we have to apply some queries to the matches retrieved. First, we remove the non-match 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, for instance, 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 Informatics of Social Security Institute (IISS), we remove the pair with the lower probability of matching as it probably is a false positive. In the case that the 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.

5 RESULTS

5.1 Experimental Setting

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

First, the Databases are stored on 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. To load the records from a CSV file (downloaded from SQL Developer) we used the Pandas Dataframe.

In terms of reduction ratio, i.e. the reduction of using Standard Blocking, instead of comparing all records of one database with the other, is always over 99%.

5.2 Results for the Quality of the Models

Before we use the models to classify records, we check the quality through the metrics: precision, recall and f-measure for both Matches and Non-Matches [Christen 2012b].

The scores for each metric have, usually, high values, from 95% to 100%. However, there are some exceptions mainly because of the quality of the data of some databases have many errors in the fields and many null values, as well.

5.3 Matching Results

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 2, the results are presented with the records to be matched between two databases and the new matches discovered . 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.

SP already linked most of the records and some databases are not supposed to link with BDIC or Immigration and Borders Service (SEF) totally. For example, the database General Statistics of Education and Science (EDUC) does not contain all the Portuguese people like BDIC, thus, those 84 968 records to match are not supposed to match with BDIC. The same happens with IEFP and General Retirement Fund (CGA).

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 records are stored into BPR.

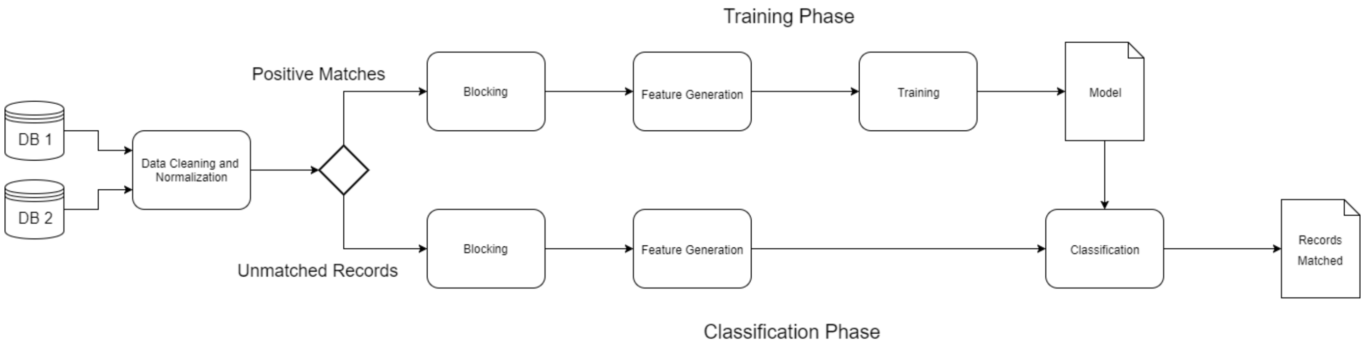

Fig. 2. Solution architecture

Table 2. 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	

6 CONCLUSIONS

This work is important because is a practical work, in an important institute, with real implications - improve the actual model of Census.

Like in every solution, the first attempt did not work perfectly but after some improvements, we had a model and a methodology that could pair two databases.

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 many unfilled 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.

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.

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.

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 for the present system.

We always used the blocking criteria based on the first name + date of birth, therefore, 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 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.

We discovered some anomalies and errors in the data like different codes, fields with non-null values that should be null or special characters. Hence, more errors should exist in the data and is necessary to find and correct them to lead to accurate results.

The current process we do is to pair two 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 with 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 have the snowball effect of finding even more matches.

Last but not least, is to have a greater automation on 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.

REFERENCES

2013. *Beyond 2011: Matching Anonymous Data.* Technical Report July. Office for National Statistics.

2016. *Metodologia de atualização da Base de População Residente - Construção da BPR 2015 (Work Document).* Technical Report. Instituto Nacional de Estatística.

C Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2, 2 (1998), 121–167. https://doi.org/10.1023/A:1009715923555

Peter Christen. 2012a. A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. *IEEE Transactions on Knowledge and Data Engineering* 24, 9 (9 2012), 1537–1555. https://doi.org/10.1109/TKDE.2011.127

Peter Christen. 2012b. *Data Matching.* Number Chapter 1. Springer Berlin Heidelberg. 1–279 pages. https://doi.org/10.1007/978-3-642-31164-2

Munir Cochinwala, Verghese Kurien, Gail Lalk, and Dennis Shasha. 2001. Efficient data reconciliation. *Information Sciences* 137, 1-4 (9 2001), 1–15. https://doi.org/10.1016/S0020-0255(00)00070-0

AnHai Doan, Alon Halevy, and Zachary âĂŐIves. 2012. *Principles of data integration.* Morgan Kaufman. https://doi.org/10.1145/2347696.2347721

Mohamed G Elfeky, Vassilios S Verykios, and Ahmed K Elmagarmid. 2002. TAILOR: a record linkage toolbox. *Proceedings 18th International Conference on Data Engineering* (2002), 17–28. https://doi.org/10.1109/ICDE.2002.994694

Christos Faloutsos and King-Ip Lin. 1995. 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* 24, 2 (1995), 163–174. https://doi.org/10.1145/223784.223812

Ivan P. Fellegi and Alan B. Sunter. 1969. A Theory for Record Linkage. *J. Amer. Statist. Assoc.* 64, 328 (12 1969), 1183. https://doi.org/10.2307/2286061

David A. Freedman. 2005. *Statistical Models: Theory and Practice.* Cambridge University Press. 414 pages. https://doi.org/10.1017/CBO9780511815867

Teuvo Kohonen. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 1 (1982), 59–69. https://doi.org/10.1007/BF00337288

Liang Jin, Chen Li, and Sharad Mehrotra. 2003. Efficient record linkage in large data sets. In *Eighth International Conference on Database Systems for Advanced Applications, 2003. (DASFAA 2003). Proceedings.* IEEE, 137–146. https://doi.org/10.1109/DASFAA.2003.1192377

J Macqueen. 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* 1, 233 (1967), 281–297. https://doi.org/citeulike-article-id:6083430

Andrew McCallum, Kamal Nigam, and Lyle L.H. Ungar. 2000. 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* (2000), 169âĂŞ178. https://doi.org/10.1145/347090.347123

Alvaro E. Monge and Charles P. Elkan. 1997. 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* (1997), 23–29. https://doi.org/10.1.1.28.8405

J. R. Quinlan. 1986. Induction of Decision Trees. *Machine Learning* 1, 1 (1986), 81–106. https://doi.org/10.1023/A:1022643204877

Sunita Sarawagi, Leo Breiman, Jerome H Friedman, A Richard, and Charles J Stone Classification. 2002. 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). https://doi.org/10.1.1.13.917

Luc Vinet and Alexei Zhedanov. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10, 8 (1966), 707–710. http://arxiv.org/abs/1011.1669http://dx.doi.org/10.1088/1751-8113/44/8/085201

J.R. Wang and S.E. Madnick. 1989. The inter-database instance identification problem in integrating autonomous systems. In *Proceedings. Fifth International Conference on Data Engineering.* IEEE Comput. Soc. Press, 46–55. https://doi.org/10.1109/ICDE.1989.47199

William E. Yancey. 2005. Evaluating string comparator performance for record linkage. *Statistical Research Division* (2005), 3905–3912. http://www.amstat.org/sections/srms/Proceedings/y2006/Files/JSM2006-000855.pdf

Su Yan, Dongwon Lee, Min-Yen Kan, and Lee C Giles. 2007. Adaptive Sorted Neighborhood Methods for Efficient Record Linkage. *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries* (2007), 185âĂŞ194. https://doi.org/10.1145/1255175.1255213

final version November 2017