APPLICATIONS OF DEEP LEARNING AND PARALLEL PROCESSING FRAMEWORKS IN DATA MATCHING

by

Bernardo Jose Najlis, Systems Analyst, Universidad CAECE, 2008

A Major Research Paper

presented to Ryerson University

in partial fulfillment of the requirements for the degree of

Masters of Data Science

In the Program of

Data Science and Analytics

Toronto, Ontario, Canada, 2018

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A MAJOR RESEARCH PAPER (MRP)

I hereby declare that I am the sole author of this Major Research Paper. This is a true copy of the MRP, including any required final revisions, as accepted by my examiners.
I authorize Ryerson University to lend this MRP to other institutions or individuals for the purpose of scholarly research
I further authorize Ryerson University to reproduce this MRP by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.
I understand that my MRP may be made electronically available to the public.

BERNARDO JOSE NAJLIS

APPLICATIONS OF DEEP LEARNING AND PARALLEL PROCESSING FRAMEWORKS IN DATA MATCHING

Bernardo Jose Najlis

Masters of Science 2018

Data Science and Analytics

Ryerson University

Most of Data Science research work assumes a clean, deduplicated dataset as a pre-condition. In reality, 80% of the time spent in data science work is dedicated to data deduplication, cleanup and wrangling. Not enough research papers focus on data preparation and quality, even though it is one of the major issues to apply Data Science. The research subject of this paper is to **improve Data Matching techniques on multiple datasets containing duplicate data using parallel programming and Deep Neural Networks. Parallel programming frameworks** (like MapReduce, Apache Spark and Apache Beam) can dramatically increase the performance of computing pair comparisons to find potential duplicate record matches, due to $O(n^2)$ complexity of the problem. **Deep Neural Networks** have shown great results to improve accuracy on many traditional machine learning applications. The problem and solution researched are of general applicability to multiple data domains (healthcare, business).

ACKNOWLEDGEMENTS

[TBD]

DEDICATION

[TBD]

TABLE OF CONTENTS

[TBD]

INTRODUCTION

[TBD]

LITERATURE REVIEW

Our literature review spanned over both academic textbooks that cover the subject of Data Matching, as well as very recent research and also reference/foundational research papers.

The main reference and consultation book used throughout our research was ***"Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution and Duplicate Detection" by Peter Christen (Springer, 2012)***. This covers all the phases on the traditional data matching approach (Data pre-processing, Indexing/Blocking, Record pair comparison, Classification, Matching, Evaluation and Clerical Review) and all traditional machine learning techniques used in each of these steps. The second major textbook used as reference was ***"Entity Resolution and Information Quality" by John R. Talburt (Elsevier, 2011)***. This book presents also a very traditional machine learning approach to the subject of Entity Resolution and provides the main concepts and principles in this research area.

As for peer reviewed research papers, they fall under two major areas:

- **Modern approaches to Entity Resolution (i.e. using parallel processing frameworks, using statistical techniques)**

    - ***"Performance Comparison of Apache Spark and Tez for Entity Resolution", Ikram Ul Haq, 2017.*** This paper applies two of the major parallel programming frameworks (Apache Spark and Tez/Hive) for Entity Resolution and compares their performance. It provides a very comprehensive and descriptive list of all technical tools and techniques used.

    - ***"FEBRL – Freely Extensible biomedical record linkage", Peter Christen – Tim Churches, 2005.*** This paper provides details on many of the traditional statistical and machine learning techniques used on Entity Resolution. It is also the project that provides the data set used in this project.

    - ***"Toward building end-to-end entity matching solutions", Paul Suganthan Gnanaprakash Christopher, 2018.*** This paper provides a very descriptive and complete dissertation on all the areas involved in the development of an entity matching solution on cloud technologies that uses a parallel programming framework.

    - ***"A Bayesian Approach to Graphical Record Linkage and De-duplication", Rebecca C. Steorts – Rob Hall – Stephen E. Fienberg, 2015.*** In this paper the authors propose the use of unsupervised machine learning techniques and graph theory to detect duplicate record probabilities and then match them.

    - ***"Efficient Parallel Set-Similarity Joins using MapReduce", Rares Vernica – Michael J. Carney – Chen Li, 2010***. This paper is the first one using the Hadoop MapReduce programming framework to the problem of Entity Resolution. It was consulted more for reference and historical value and as a stepping stone to understand the application of parallelization applied to ER.

- ***"Duplicate Detection on GPUs", Benedikt Forchhammer - Thorsten Papenbrock - Thomas Stening - Sven Viehmeier - Uwe Draisbach - Felix Naumann, 2011.*** This paper presents the novel approach of using GPUs (graphical processing units) as a mean to increase efficiency when computing entity resolution in large data sets, due to the $O(n^2)$ nature of the problem.

- ***"Parallel Entity Resolution with Dedoop", Lars Kolb – Erhard Rahm, 2012.*** This is the first most popular paper to apply the use of parallel programming frameworks (Hadoop MapReduce) in the area of Entity Resolution.

- ***"Data Partitioning for Parallel Entity Matching", Toralf Kirsten - Lars Kolb, Michael Hartung - Anika Groß - Hanna Köpcke - Erhard Rahm, 2010***. This paper focuses on the sub-area of data partitioning (to reduce the $O(n^2)$ nature of the Entity Resolution problem). Even though it is a bit dated (all programming was done using Java in the pre-Hadoop era), concepts and ideas for this are still valid to understand the value of partitioning.

- ***"Learning Blocking Schemes for Record Linkage", Matthew Michelson – Craig A. Knoblock, 2006.*** This paper presents the novel approach of using statistical learning techniques to infer partitioning/blocking sets from the data itself.

- ***"Record Linkage", Atasha Ann Bown Larsen, 2013***. This thesis reviews the application of different blocking techniques to different series of datasets to find optimal results using data and concepts from the FEBRL paper.

- ***"An efficient spark-based adaptive windowing for entity matching", Demetrio Gomes Mestre - Carlos Eduardo Santos Pires - Dimas Cassimiro Nascimento - Andreza Raquel Monteiro de Queiroz - Veruska Borges Santos - Tiago Brasileiro Araujo, 2017***. This is one of the most recent papers in the subject on Entity Resolution, elaborating on the importance of parallel processing for entity resolution on large data volumes, and specifically focused on spark cluster nodes workload distribution aimed for this type of problem.

- **Deep Neural Networks**

  - ***"Learning Text Similarity with Siamese Recurrent Networks", Paul Neculoiu – Maarten Versteegh – Mihai Rotaru, 2016.*** This paper explores the use of deep neural networks to obtain text similarity metrics. It uses bi-directional LSTMs (Long Short Term Memory) with a Siamese architecture.

  - ***"Siamese Recurrent Architectures for Learning Sentence Similarity", Jonas Mueller – Aditya Thyagarajan, 2016.*** This paper also focuses on the use of LSTM networks for labeled data comprised of pairs of variable-length sequences, that assesses semantic

similarity between sentences.

- ***"Long Short-Term Memory", Sepp Hochreiter – Jurgen Schmidhuber, 1997.*** This is the foundational paper on LSTM, so it was covered and studied for reference purposes.

- ***"Learning to Forget: Continual Prediction with LSTM", Felix A. Gers - Jurgen Schmidhuber – Fred Cummins, 2000.*** Another foundational paper in the subject of LSTMs, consulted mainly for reference purposes in the understanding of Long Short-Term Memory networks.

EXPLORATORY DATA ANALYSIS

The datasets were obtained from one of the most popular Record Linkage projects, referenced by almost every other paper reviewed: FEBRL (Freely Extensible Biomedical Record Linkage). The complete dataset is composed of several files: as the goal is to find linkage/duplicates between two sets, there are two subsets (dataset A and dataset C) and also there are 4 groups with different numbers of records each (1,000, 2,500, 5,000 and 10,000). Each file is in CSV (comma separated value) text format and includes the column names header in the first row. All files have the same schema (same number of columns and at the same position).

Here is a description of the file schema, that can serve as data dictionary:

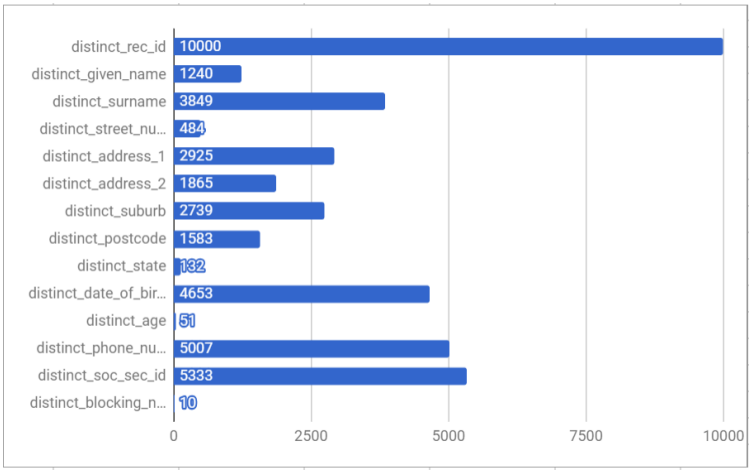
Col Name/ Feature	Description	Data Type	Null / Blanks	Min	Max	Average
rec_id	Unique identifier for each row	Integer	NO	0	49,991	24,995.5
given_name	First name of the patient.	String	YES	N/A	N/A	N/A
surname	Last name of the patient.	String	NO	N/A	N/A	N/A
street_number	Address number of Patient's address.	Integer	YES	0	12,999	77.18
address_1	Street name of patient's address.	String	NO	N/A	N/A	N/A
address_2	Additional information (apt number, others) of patient's address.	String	YES	N/A	N/A	N/A
suburb	Suburb name of patient's address.	String	YES	N/A	N/A	N/A
postcode	Postal Code of patient's address.	Integer	YES			
state	State name of patient's address.	String	YES	N/A	N/A	N/A
date_of_birth	Date of birth of patient. Codified as integer (4 first digits correspond to year, next two correspond to month, last two correspond to day). Months and days have leading zeroes if applies.	Integer	YES	18971218	19999017	19491159
age	Age of patient.	Integer	YES	2	98	29.16
phone_number	Phone number of patient. Format is: two digits for area code (with leading zeroes) and 8 digits for phone number.	String	YES	N/A	N/A	N/A
soc_sec_id	Social Security ID for patient. 7 digits	Integer	NO	1000606	9998137	5531936.39

blocking_number	Number of block assigned for comparison. This can be used as a label when applying blocking strategies.	Integer / Categorical	NO	0	9	4.5

To understand the nature of the dataset we will be working with during our project, we performed exploratory data analysis on the 10,000 records dataset.

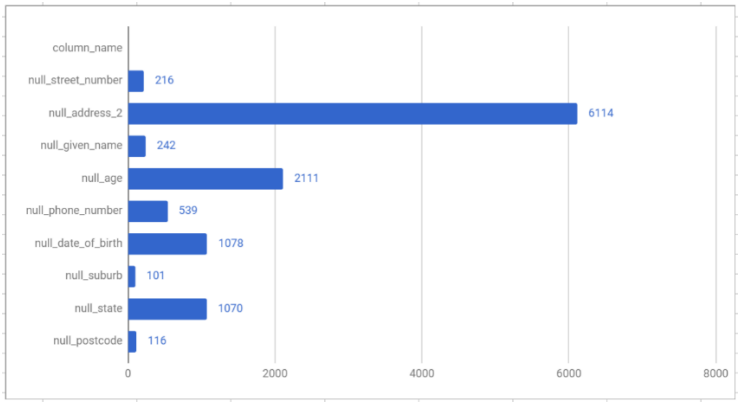
As the purpose of this dataset is to find duplicates within the rows (patients), there is no sense on charting bi-variate exploratory data analysis: correlations between variables are of no use to solve our problem. All of the exploratory data analysis charts are based on the nature of the dataset (number of rows/patients per distinct features) and their distributions.

Number of distinct values per feature



All records have a unique rec_id (as expected) and there are about 5,000 duplicate phone_numbers and soc_sec_id. This means those two variables are great candidates to be used as unique identifiers, together with date_of_birth (4,653 unique values) and surname (3,849 unique values).

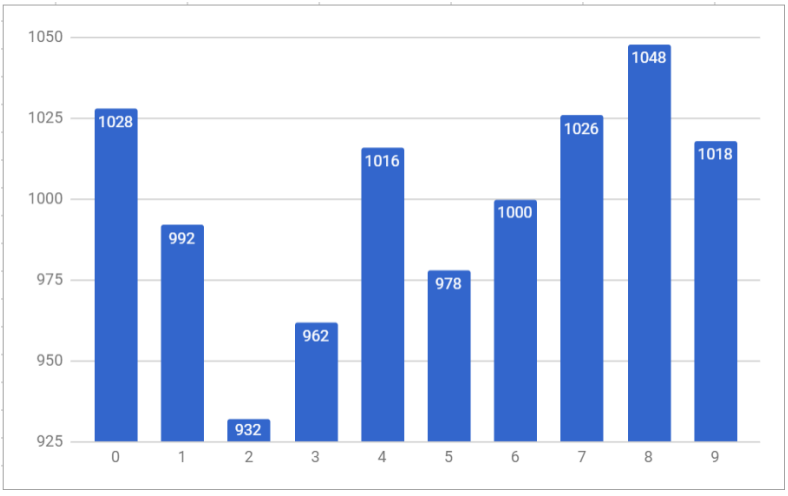
Number of null / missing values per feature (only for features that are nullable)



More than half of the records do not have address_2 data, and about 20% of the dataset has no age (although some of them may be inferred from the date of birth, as only 1,078 rows do not have dob).

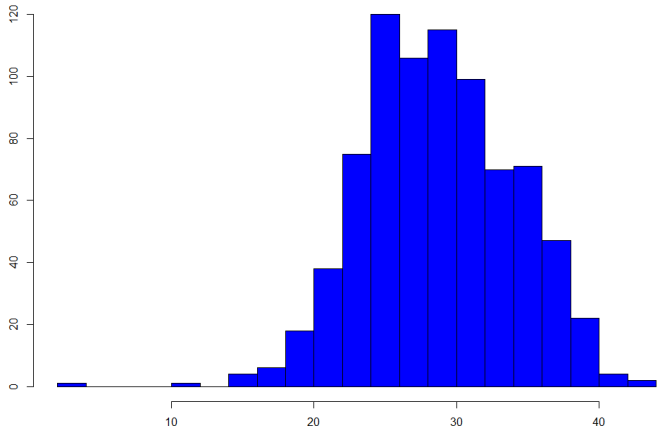
All other fields are at less than 10% of missing / null values data.

Number of patients per blocking number



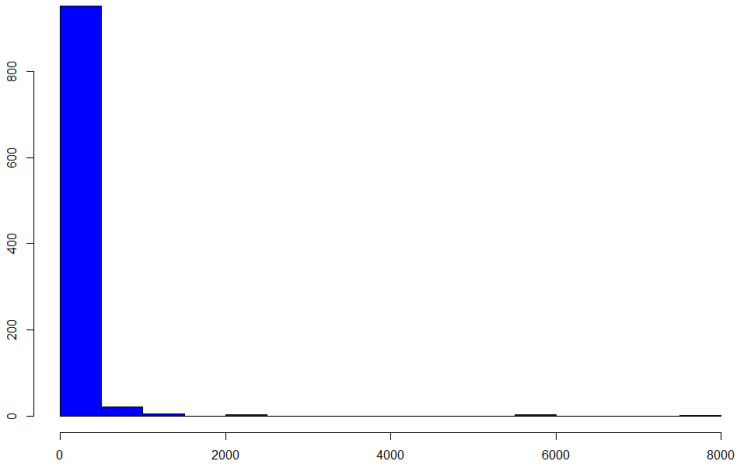
There are 10 blocks (0 through 9) and all have about 1,000 records (max: 1048, min: 932) so we can consider this as a fairly balanced distribution set.

Number of patients per age



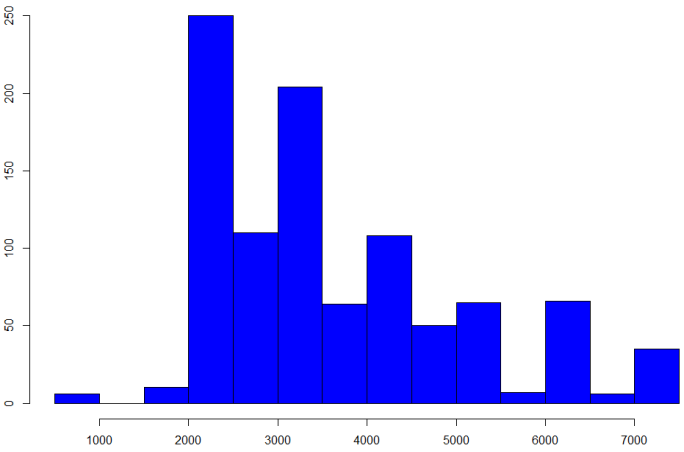
All patients have ages in a reasonable range (min: 2, max: 98) with an average of 29.16 years old. As we can see in the histogram, this is a normally distributed set.

Number of patients per Street Number



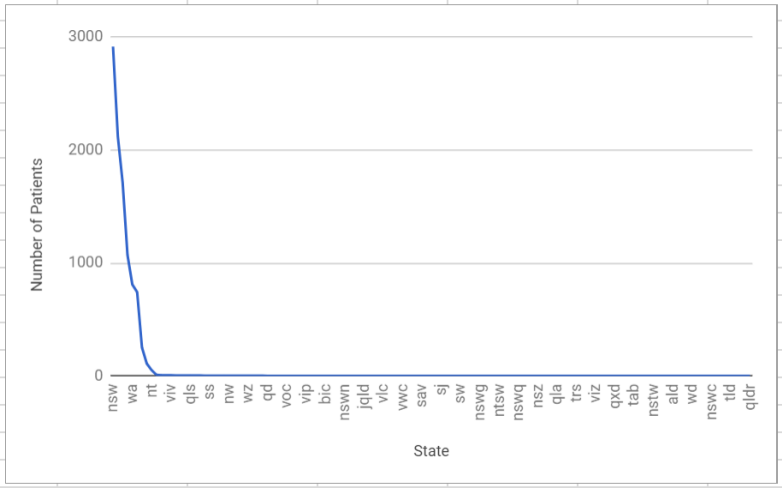
Although street number is not a traditional continuous variable, we decided to chart its histogram to understand if there is any correlation or bias in its distribution. Most of all address numbers have numbers less than 500, which indicates a very unbalanced set on this variable.

Number of patients per Postal Code



Same as street number, postal code is more a categorical variable expressed with integers than a conventional continuous variable. Still its histogram shows a quasi-normalized distribution slightly skewed to the left.

Number of patients per State



Same as with other variables, state is more a descriptive / categorical value. Its histogram shows that more than half of patients are located in 3 states, as the rest follows a very long tail distribution.