Evaluation of Machine Learning Algorithms in a Human-Computer Hybrid Record Linkage System

Mahin Ramezani[a,b], Guru Ilangovan[a,b] and Hye-Chung Kum[a,b]

[a]*Department of Computer Science, Texas A&M University, 400 Bizzell St, College Station, TX 77843, USA*
[b]*Population Informatics Lab, Department of Health Policy and Management, Texas A&M University School of Public Health, 212 Adriance Lab Rd, College Station, TX 77843, USA*

Abstract

Record linkage, often called entity resolution or de-duplication, refers to identifying the same entities across one or more databases. As the amount of data that is generated grows at an exponential rate, it becomes increasingly important to be able to integrate data from several sources to perform richer analysis. In this paper, we present an open source comprehensive end to end hybrid record linkage framework that combines the automatic and manual review process. Using this framework, we train several models based on different machine learning algorithms such as random forests, linear SVM, Radial SVM, and Dense Neural Networks and compare the effectiveness and efficiency of these models for record linkage in different settings. We evaluate model performance based on Recall, F1-score (quality of linkages) and number of uncertain pairs which is the number of pairs that need manual review. We also test our trained models in a new dataset to test how different trained models transfer to a new setting. The RF, linear SVM and radial SVM models transfer much better compared to the DNN. Finally, we study the effect of name2vec (n2v) feature, a letter embedding in names, on model performance. Using n2v results in smaller manual review set with slightly less F1-score. Overall the SVM models performed best in all experiments.

Keywords

Record Linkage, deduplication, entity resolution, machine learning, Benchmarking, patient matching

1. Introduction

As the amount of data that is generated grows at an exponential rate, it becomes increasingly important to be able to integrate data from several sources to perform richer analyses. For example, the research on covid can be accelerated if all fragmented patient data could be integrated. In error-free clean databases with unique identifiers common to all the databases, integrating them can be easily accomplished with simple joins[1]. However, such identifiers are often not available in real world data. In that case, the available fields common to the databases are compared and

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021) - Stanford University, Palo Alto, California, USA, March 22-24, 2021.
mahin@tamu.edu (M. Ramezani); ilan50_guru@tamu.edu (G. Ilangovan); kum@tamu.edu (H. Kum)
https://pinformatics.org/ (H. Kum)

a decision has to be made on whether the two records refer to the same real world entity or not. This problem of finding data records in heterogeneous databases that refer to the same entities is referred to as *record linkage (RL)* or *entity resolution*. When finding data records for the same entities in one database, this problem is also called *de-duplication* for linking the database to itself.

Automated record linkage methods have been studied extensively in many fields since the problem was first introduced by Newcomb[2]. The best results may be obtained by a hybrid human-computer linkage process that augments the results of automatic algorithms with human judgement[3]. It involves a small team of well trained human experts reviewing potential uncertain pairs generated by algorithms and first making independent decisions then comparing notes on disagreements and coming to consensus [4, 5, 6]. Probabilistic methods and rule-based approaches were the most common automated approaches but machine learning (ML) approaches are rapidly gaining traction and proving to be the preferred automatic linkage methods.

In this paper, we present a comprehensive end to end hybrid record linkage framework that combines the manual review and the automated process to achieve both scalability and high quality linkage results. Quality control in any record linkage project is critical because all approaches will result in some level of incorrect matches that will generate erroneous integrated data as well as miss correct matches resulting in a fragmented integrated dataset. We achieve the best of both worlds by allowing the automated algorithms to resolve majority of the linkages that have a high probability of being either a match or non-match, but also have the option to send ambiguous pairs to human experts for final determination to improve the linkage quality[7]. Hence, the goals of this hybrid record linkage process is to achieve optimum linkage quality, both in terms of no mismatches and no true matches missed, while still minimizing the amount of manual review required to achieve this quality. This paper focuses on comparing how well different ML algorithms meet this goal. We also investigate how well different ML models trained on one dataset transfer to other settings within the USA. Determining which ML models transfer better to other settings is important because one of the difficulties to using ML methods on real projects is challenges to building a training set that is comprehensive enough to build good models. In addition, we studied how adding letter embedding[8] in names will effect the performance of these models. In sum, the contributions of this paper are:

- A hybrid open source RL framework that can achieve scalability and high quality results
- A comparison of four different RL ML algorithms in meeting the goals of the hybrid system
- A comparison of how well ML RL models trained on one dataset transfer to different settings
- An evaluation on the impact of using letter embedding in names in RL ML algorithms

The rest of the paper is laid out as follows. In section 2, we briefly review the RL literature on using ML algorithms. Section 3 describes a hybrid record linkage framework and section 4 describes the experimental design of our evaluation. Section 5 and 6 then describes the results from the individual experiments and discuss main insights. Finally, Section 7 presents our conclusions.

2. Related Works

Use of ML algorithms for RL has been studied in different contexts. For instance, a radial basis kernel SVM was used successfully to link genealogy records from 19th century Canada[9]. Random forests for RL in financial entity recognition[10] and author disambiguation[11] demonstrated the efficiency of random forests for this task. With the recent re-emergence of neural networks, a lot of research shows the potential for neural networks in entity resolution. Using structured neural networks for genealogical RL was shown to give very reliable results[12, 13]. Feigenbaum[14] compared the performance of ML algorithms (SVM, random forest), logistic regression, and other heuristic approaches on US census dataset and showed that SVM did slightly better than RF in both true positive rate (TPR) and positive predictive value (PPV). Ilangovan[15] studied the effectiveness and efficiency of different ML algorithms (SVM, Random Forest, and neural networks) in a controlled experiment with different level of heterogeneity in data and size of training dataset. He found that RF and SVM performed very well both in terms of traditional metrics like F1 score as well as manual review set size for error rates from 0% to 60%. In [16] the performance of SVM and Random Forest was compared to investigate the upper bound achieved in the linkage rate and the conditions required to achieve the rate. The results of this study illustrated that the RF produced high quality result at threshold value >= 0.85 while for the cases where quantity is the main concern SVM with a lower threshold is recommended.

3. Method

In the health sector, incorrectly linking records that belong to different patients with similar identifiers such as twins or family members may lead to serious harm due to incorrect health information (e.g., medication allergies). Thus, often algorithms are tuned to minimize false matches which inevitably increases the rate of missing true matches leaving the health records fragmented. This also lead to its own problems (e.g., incomplete medical history). More problematic is that the unlinked true matches are often biased because there are more issues with identifying information in lower socioeconomic populations such as ethnic names[17]. On the other hand, manual RL methods may be prohibitively time-consuming. One solution is to use a hybrid record linkage framework which combines the automated process and the manual process (see Figure 1). First, the automated algorithm will handle the records that have high probability of either a match or unmatch, which for most applications is majority of the data, and then a human expert will resolve those remaining records that the algorithms were uncertain on. Thus, in automated record linkage algorithms one threshold is used to divide the data into two classes (match, unmatch), while hybrid methods use two thresholds to form three classes (match, uncertain, unmatch).

3.1. Pair generation

The first step in hybrid RL process is the creation of pairs from one or more databases for potential matches such as those that share some common identifier. Often referred to as *blocking*, the idea

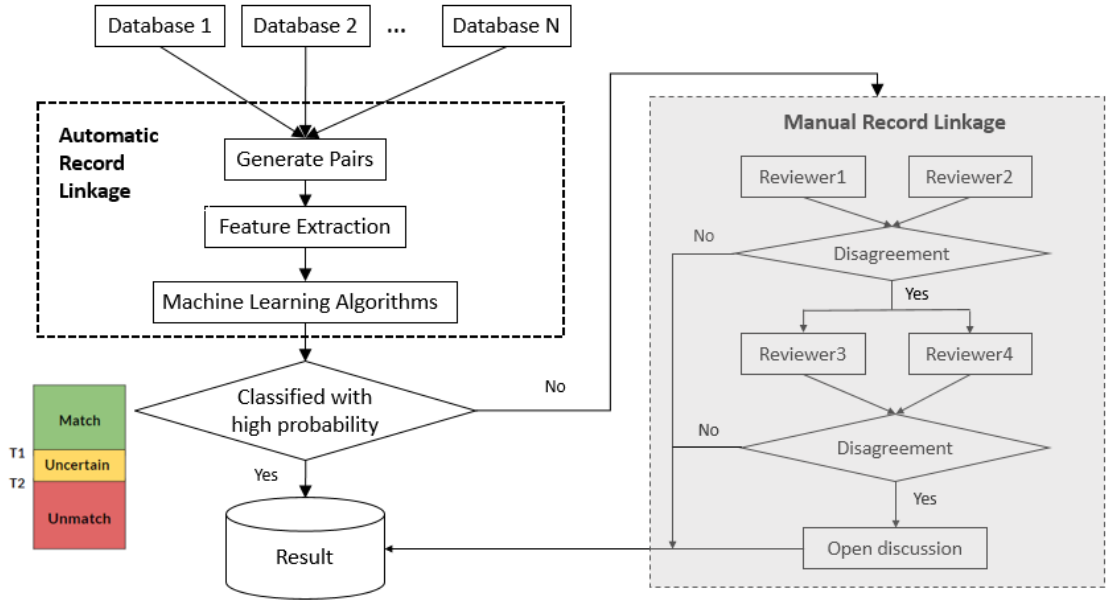

Figure 1: Hybrid Record Linkage framework

is to use the identifier field just to generate candidate potential pairs to reduce computation. Once the pairs are generated, features would be extracted from each pair and fed into the ML models. In this study, we generated the pairs files by blocking on appropriate fields. For example, first name and last name, first name and date of birth, last name and date of birth, etc.

3.2. Feature extraction in pairs

After generating the pairs, we need to extract some useful features to feed them to ML algorithms. For each pair of first and last names, we calculated the Jaro Winkler (JW) distance, Damerau–Levenshtein (dl) distance, Longest Common Substitution (LCS) distance, and Name2Vec (n2v) distance[8] which is a name-embedding using Doc2Vec methodology, where each name is a document and each letter of the name is considered a word. We trained two separate models using names from two public US datasets. Using code from[8] and all names extracted from the North Carolina Voter Registration data[1] and ONC Patient Matching Algorithm Challenge data[2], we trained separate models for first name and last name. Using these models, we calculated n2v distances between each names in pairs. We also created a boolean feature to detect when first name and last name were swapped. Finally, the normalized frequency of the first and last names

[1]https://www.ncsbe.gov/results-data/voter-registration-data
[2]https://linkagelibrary.icpsr.umich.edu/linkagelibrary/project/111962

in their respective databases was also added to capture how rare the name is.

For each pair of dates of birth, we calculated the DL distance, the DL distance for the year, month, and day components individually, and a boolean feature to detect when month and day were swapped. Finally, we used the raw birth years as a feature for age. Beside names and date of birth features, DL and LCS distances have been calculated for address, phone and SSN. We also engineered a binary feature to capture possible last name change due to marriage for females over eighteen because this is one of the common issues in RL in the USA. Finally, we coded gender as three categories based on the gender of the pair as ff, mm, or different.

3.3. Machine Learning algorithms

After extracting the features, the next step is to train the ML algorithms using those features. For this study, we have used four ML algorithms to generate different models: Random Forest, Linear and Radial Support Vector Machine, and Dense Neural Network.

3.3.1. Random Forest

In order to train a random forest model, a grid search with 10-fold cross-validation was used on the training set to tune the maximum number of features at each split hyper-parameter, tested for 3, 5, 7, 9, 11, 13 and 15. As the number of estimators goes up, the performance typically goes up initially and plateaus after a point. The performance of the random forest started plateauing at about 250 estimators. Thus, the number of estimators was fixed at 350, which allowed a margin of 100 to ensure optimal performance. Once the best performing hyper-parameter was identified, the random forest model was rebuilt on all of the training data using the same hyper-parameter.

3.3.2. Radial SVM and Linear SVM

Two support vector machines were built, one with a radial basis function and the other with a linear basis function. The two key parameters for a radial basis kernel SVM were the penalty parameter, C and the kernel coefficient, sigma. Those two parameters were tuned using 10 fold cross-validation and grid search on the training data. The grid was validated for all combinations of C (0.1, 0.5, 1, 10) and sigma (0.03, 0.5, 0.9). The penalty parameter, C, was trained similarly for the linear SVM. The models were retrained on all of the training data once the hyper-parameters were fixed.

3.3.3. Dense Neural Network

The neural net had one input layer, two hidden layers and one output layer with dense connections between all layers. The input layer had 33 units (from the feature vector discussed in section 3.2). The two hidden layers had 64 units each with relu activation functions. After the first layer, there was batch normalization and 0.1% dropout. There was a batch normalization after the second layer but no dropout as is the standard practice for layers preceding the output layer. The output

layer had 1 unit with a sigmoid activation that returned the probability of a match. An RMSprop optimizer with a learning rate of 0.001 was used with binary cross-entropy as the loss function. The batch-size was maintained at the default 32 and the network was trained for 20 epochs. 20% of the data provided was used as the validation set for monitoring the validation performance.

4. Experimental Design

4.1. Data

In this study, we use two different real world datasets. The first is a large gold standard RL dataset to train the ML models[18], then we evaluate how well the models transfer to another dataset.

4.1.1. Hospital EHR data

For training our models, we used a large academic hospital EHR data. This dataset is based on 10,000,000 pairs that were generated from a hospital EHR dataset by blocking on first name and last name, first name and date of birth, last name and date of birth, and social security number. A gold standard dataset[18] was developed by randomly selecting 20,000 pairs and then reviewed by consensus among a team of 4 people through independent review. In our study, we randomly split the gold standard data into 10,000 for training data and 10,000 for test data. The test data had a total of 613 linkages that needed to be identified. 55% of the records were female, 44% male, and the remaining 1% was undesignated.

4.1.2. NC voter registry data

The North Carolina State Board of Elections curates large amounts of data on state elections and voter registration. With several exceptions, this data is public and can be downloaded from https://www.ncsbe.gov/results-data/voter-registration-data. We link data from two time points (May 2017 and July 2020) using the voter registry number as the gold standard. Note that this dataset is only used to test the trained models. We perturb this data by randomly generating month and day of birth to account for twins as in [15]. Using the records from Yancey county, we generated 10,000 pairs by blocking on first name and last name, first name and date of birth, and last name and date of birth. There were a total of 1773 linkages that needed to be identified. There were 12700 unique records from which 52.8% of the records were female, 46.1% male, and the remaining 1.1% was undesignated. Since the data was from a voter registry, the minimum age on the month of data pull was 18 years. People with age 65 or above formed the biggest chunk of the records followed by middle aged populations (45 to 65).

4.2. Evaluation criteria

For evaluating the models we used three measures: (1) the number of pairs that need manual review, (2) F1-score for automated results only, and (3) Recall over all results. In the hybrid RL

system, the number of pairs that need manual review is determined by two thresholds, T1 and T2, that are used to determine uncertain pairs. In this study, we defined T1 and T2 in terms of Positive Predictive Value (PPV) and Negative Predictive Value (NPV) which are calculated as in equation (1) where TP is the True Positive, TN is the True Negative, FP is the False Positive and FN is the False Negative. Since accurate results are very important in the health domain, we selected T1 and T2 such that among all predictions with probabilities above and below them respectively, the predictions were perfect on the training data.

$$PPV = \frac{TP}{TP + FP} \qquad NPV = \frac{TN}{TN + FN} \tag{1}$$

F1-score is a common measure of linkage quality. It is the harmonic mean between precision and recall (equation (2)) and hence is very useful in evaluating the effectiveness of linkage balancing between false positives and missing true links. We use the F1-score for ML RL models ($F1_{autoRL}$), which focuses on the effectiveness of only the subset of pairs that are labeled using the automated methods outside T1 and T2. In addition, we use $Recall_{overall}$, which takes into account the full set of pairs of the whole hybrid system and depicts how much of the overall TP have been correctly identified by the automated methods. If the $Recall_{overall}$ is too low, that means the automated methods are mainly good at reducing the manual review work and rely mostly on the manual process to detect much of the correct links and is roughly correlated with the manual review set size. On the other hand, very high $Recall_{overall}$ shows that the ML models captured most of the linkage and we may not need to spend much time for manual review.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \qquad F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2}$$

4.3. Study design

Three experiments have been designed to answer three study questions:

1. How well do the four ML RL algorithms meet the two goals of the hybrid RL system?
2. How well do the four ML RL models trained on one dataset transfer to different settings?
3. How does adding the n2v feature affect the different results in the experiments?

In the first experiment, we used only the hospital gold standard data. We trained using the 10,000 pairs in the training dataset then evaluated the performance on the other 10,000 testing data. For each pair, we calculated the 33 features described above (excluding the two n2v features). In the training phase, 9000 pairs were used for training and 1000 pairs were used as the validation set to tune the hyper-parameters. After hyper-parameter tuning, we trained the models with the selected hyper-parameter one more time on the whole training set. In our second experiment, we test the trained models from previous experiment on 10,000 pairs randomly generated from the NC voter dataset to see how well different ML models transfer to different settings. We ran this experiment 100 times and report the mean and standard deviation for each model (Table 2). Finally, as our last experiment, the first and the second experiments were repeated but this time

Table 1
Comparing the performance of four ML models.

Experiment	Model	manual review	$Recall_{overall}$	$F1_{autoRL}$	TP	FP	TN	FN	Total
Exp1: EHR data 613 linkages	RF	306	0.625	0.992	383	1	9305	5	9694
	Radial SVM	187	0.721	0.98	442	2	9353	16	9813
	Linear SVM	321	0.612	0.991	375	0	9297	7	9679
	DNN	217	0.936	0.989	574	11	9196	2	9783
Exp2: voter data 1773 linkages	RF	4479	0.197	0.987	354	1	5158	8	5521
	Radial SVM	1857	0.815	0.986	1463	7	6637	36	8143
	Linear SVM	1553	0.589	0.991	1057	1	7371	18	8447
	DNN	1836	0.175	0.198	315	1393	5295	1161	8164
Exp3a: EHR data 613 linkages	RF + n2v	75	0.962	0.985	590	15	9317	3	9925
	Radial SVM + n2v	34	0.956	0.975	586	16	9350	14	9966
	Linear SVM + n2v	64	0.967	0.977	593	24	9315	4	9936
	DNN + n2v	239	0.93	0.99	570	10	9180	1	9761
Exp3b: voter data 1773 linkages	RF + n2v	4386	0.215	0.988	386	1	5219	8	5614
	Radial SVM + n2v	927	0.773	0.983	1388	2	7638	45	9073
	Linear SVM + n2v	1689	0.33	0.98	594	0	7693	24	8311
	DNN + n2v	3015	0.172	0.213	309	1329	4391	956	6985

the two n2v features, one feature for first name and one for last name, were added. Thus, in this experiment, each record had 35 features. The main purpose of this experiment was to observe the effectiveness of the n2v distance on the hybrid record linkage process.

5. Results

5.1. The performance of different ML models

Table 1-Exp1 compares the four algorithms. Both training and test datasets are from the hospital data. Results demonstrate that although $F1_{autoRL}$ scores were comparable across all methods, random forest and linear SVM did worst in the $Recall_{overall}$ at slightly over 60%. Close to 40% of the linkages has to be found through manual review. This is consistent with the considerably bigger manual review size for these two algorithms compared to Radial SVM and DNN.

5.2. The performance of different ML models on a new setting

Experiment 2 studied how well the ML models trained in one setting transfers to different data. We used the trained model that we created using the EHR data to test those models on a different dataset (Voter Registry data). As seen in Table 1-Exp2, the two SVM models transfer to the new setting best, followed by RF, then DNN. Although the manual review set size are bigger than previous experiments, the $F1_{autoRL}$ is still reasonable in the three models which is important because the manual phase can then fill in the gap even if $Recall_{overall}$ is somewhat low. DNN model is not usable because there are too many FP and FN, that cannot be overcome in the manual review phase. Table 2 shows the mean and standard deviation of the 100 repeated experiments.

Table 2
Mean and standard deviation for 100 runs on voter data.

	Model	manual review size	$Recall_{overall}$	$F1_{autoRL}$
(a)	RF	4459 ± 43.5	0.193 ± 0.00768	0.982 ± 0.00443
	Radial SVM	1841.5 ± 32.5	0.81 ± 0.00809	0.983 ± 0.00192
	Linear SVM	1546.9 ± 29.2	0.58 ± 0.00968	0.989 ± 0.00181
	DNN	1827.6 ± 29.6	0.171 ± 0.00822	0.194 ± 0.00918
(b)	RF + n2v	4378.4 ± 41.5	0.212 ± 0.00789	0.986 ± 0.0032
	Radial SVM + n2v	922.8 ± 27.2	0.769 ± 0.0088	0.983 ± 0.00196
	Linear SVM + n2v	1662.1 ± 33.5	0.334 ± 0.00898	0.976 ± 0.0035
	DNN + n2v	2997 ± 38.9	0.163 ± 0.00831	0.203 ± 0.00997

5.3. Effect of n2v feature on model performance

For the last experiment, we added 2 more features: n2v distance for first name and last name to see how adding these features can change the results in the first and the second experiments. As seen in Table 1-Exp3a, adding n2v caused a big reduction in the number of pairs that needed manual review, significant improvements to $Recall_{overall}$, with only slight change in $F1_{autoRL}$ for all models except DNN. Results were somewhat similar for the second experiment presented in Table 1-Exp3b, although the impact on $Recall_{overall}$ was not seen. It seems that adding n2v, increases the confidence (higher probabilities) of the model predictions which means potentially more matches and unmatches are detected through automated step, and the number of observation classified as uncertain is smaller. However in the DNN model the impact was opposite with higher manual review size and slightly higher F1-scores. Figure 2 depicts the effect of adding n2v features to ML model performance in the first experiment.

6. Discussion

This research sought to systematically study the performance of the different ML algorithms (Random Forest, Linear SVM, Radial SVM, and Dense Neural Network) on different settings for a hybrid record linkage method in terms of F1 score, Recall, and size of manual review. The automatic ML based RL code and models can be downloaded from https://github.com/pinformatics/hybridRL_code_and_models. Users can use the trained models to conduct record linkage on their data or train a new model using their own dataset.

In the first experiment, the most interesting finding was that although RF had the best $F1_{autoRL}$ score, it was not a good model overall for the hybrid system. The manual review size for radial SVM was only 61% of random forest model at the cost of having more false labels, but little impact of $F1_{autoRL}$ scores. Radial SVM missed 16 true matches while DNN had 11 false links. And both these models were able to identify many more of the linkages, giving much better $Recall_{overall}$ scores. It is very clear that in a hybrid system the price for perfect performance in the first pass has to be paid by a lot of manual review in the second pass. Obviously, the performance requirements for the algorithms are affected by the importance of the linkage task. When medical databases

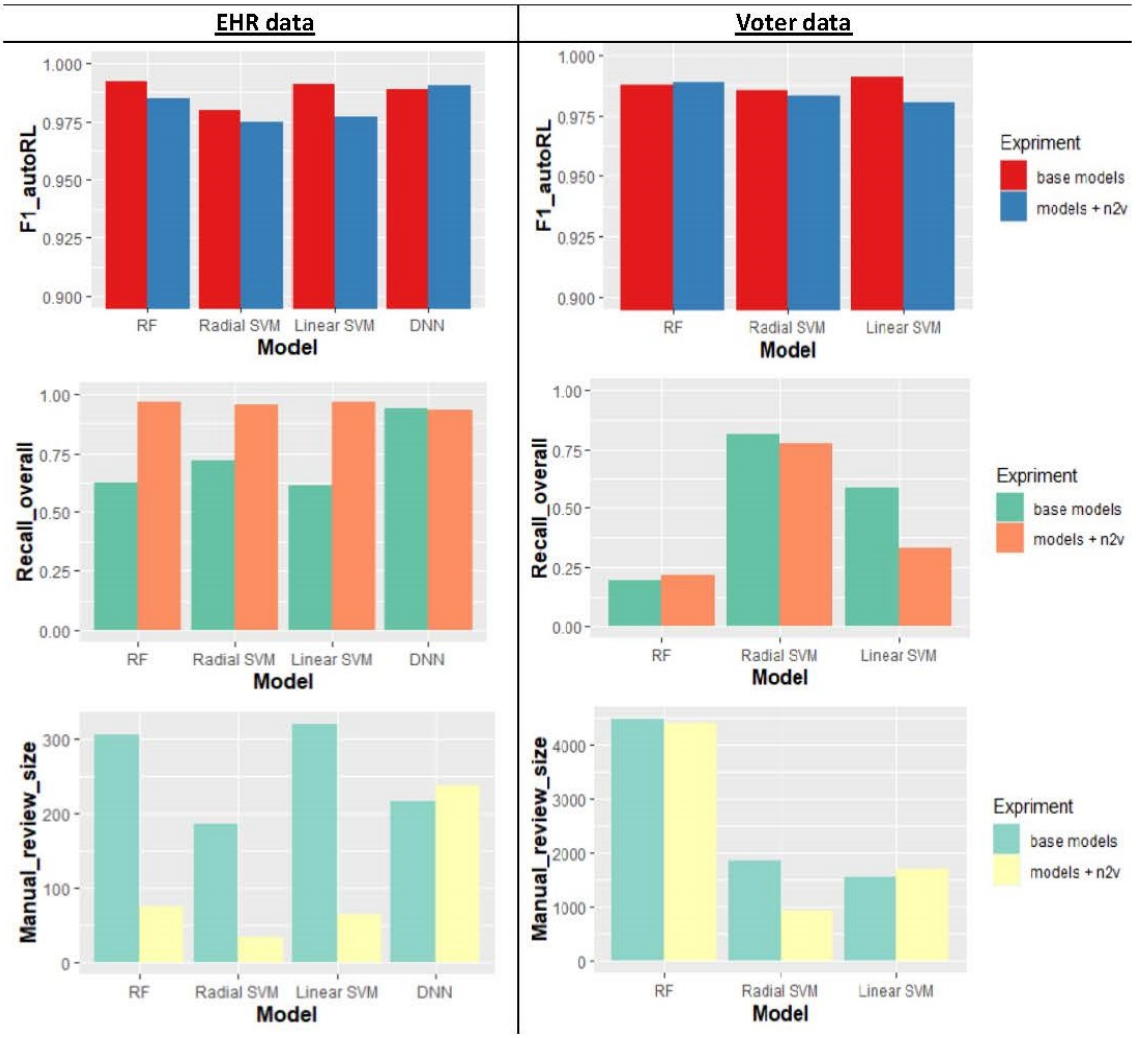

Figure 2: Comparing the performance of different ML RL models.

are to be linked[18], the process is often critical and requires thresholds that give perfect (100%) results in the training set. However, in some domains like genealogy records, small error rates are often acceptable. In that case, users can relax the thresholds to meet project requirements that can result in smaller manual review set. Thus, depending on the performance requirements of the task at hand, the performance requirements for the automatic linkage can be defined. This can potentially save a lot of time and effort on the manual linkage.

The results of the second experiment demonstrate that RF, linear SVM and radial SVM mod-

els transfer to a new setting much better than DNN. For these three models, $F1_{autoRL}$ score was comparable, however the manual review size is much bigger than previous experiments and impact the $Recall_{overall}$ score. The results indicate that the models built on EHR data can be used to identify clear non-matches, and identify some number of true matches but manual review is required to identify most of the true matches. SVM models perform much better than the RF by reducing the manual review size to less than 42% while also identifying over 60% of the linkages. In comparison, RF model had over 40% manual review size and only 20% of linkages. DNN performance was not acceptable to be used and seems to indicate that the model may be over fitting to the data it was trained on.

The goal of the last experiment was to see how adding the n2v features, which is a letter embedding for names, can affect the results. Clearly adding n2v features can reduce the size of manual review significantly on all models except DNN in the first experiment (Table 1-Exp3a). As expected, the results indicate that using n2v distance for first name and last name is similar to the impact of adding in approximate name matching where it can increase identification of true linkages automatically but at the cost of also increasing false linkages. More concretely, $Recall_{overall}$ improved noticeably to over 90% when n2v features were added, but at the same time number of FP went up from 1,2,0 to 15,16,24 respectively for RF, Radial SVM, and Linear SVM. Remember that these errors cannot be corrected during the manual review phase, so the chose of using n2v or not will depend on the error rate that is acceptable in the given application. Adding n2v features had little impact on the DNN model performance which was low anyway.

The impact of adding n2v features to model performance applied to a different dataset (voter data) in Table 1-Exp3b was seen most in the two SVM models. Both RF and DNN had comparable low results on all measures except DNN that has increased the manual review size making things worse. In comparison, radial SVM models reduced the manual review size by more than half (1857 to 927). The interesting finding was that this reduction did not translate directly into improvements in $Recall_{overall}$ where radial SVM had a significant reduction to 77%. Upon closer look, we can see that most of the reduction in manual review was due to pairs that were confirmed correctly as TN in the radial SVM model. In comparison, the linear SVM dropped many TP (from 1057 to 594) when n2v was added reducing $Recall_{overall}$. Thus, the radial SVM model benefited the most from adding in the n2v features with negligible impact on Recall and F1 score.

7. Conclusion and future works

Automatic record linkage methods have made significant progress during the last few decades, however they still may not have the high degree of reliability of manual record linkage. On the other hand, the manual record linkage method is very expensive and time-consuming. Thus, in this paper, we presented and evaluated an open source hybrid record linkage framework that combines the manual process and the automated process to achieve both scalability and high quality linkage results. More work is needed to test if more complex and effective neural network models may have better performance. In addition, future work is needed to systematically quantify the

biases in RL by race to study the impact of RL on health disparities database studies.

References

[1] E. Fosbøl, et al., Prehospital system delay in st-segment elevation myocardial infarction care: A novel linkage of emergency med. svcs in hospital registry data, AHJ 165 (2013) 363–370.

[2] Newcombe, et al., Automatic linkage of vital records, Science 130 (1959) 954–959.

[3] M. Karim, et al., View: a framework for organization level interactive record linkage to support reproducible data science, 2021. arXiv:2102.08273.

[4] H.-C. Kum, et al., Enhancing privacy through an interactive on-demand incremental information disclosure interface: applying privacy-by-design to rl, in: {SOUPS}, 2019.

[5] E. D. Ragan, H.-C. Kum, G. Ilangovan, H. Wang, Balancing privacy and information disclosure in interactive record linkage with visual masking, in: SGICHI 2018, 2018, pp. 1–12.

[6] M. e. a. Bailey, How Well Do Automated Linking Methods Perform in Historical Data? Evidence from New US Ground Truth., Technical Report, Mimeo, 2018.

[7] H.-C. Kum, A. Krishnamurthy, A. Machanavajjhala, M. K. Reiter, S. Ahalt, Privacy preserving interactive record linkage (ppirl), JAMIA 21 (2014) 212–220.

[8] J. Foxcroft, A. d'Alessandro, L. Antonie, Name2vec: Personal names embeddings, in: Canadian Conference on Artificial Intelligence, Springer, 2019, pp. 505–510.

[9] B. E. Mumma, D. B. Diercks, B. Danielsen, J. F. Holmes, Probabilistic linkage of prehospital & outcomes data in out-of-hospital cardiac arrest, Prehospital Emerg. Care 19 (2015) 358–364.

[10] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, in: Soviet physics doklady, volume 10, 1966, pp. 707–710.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al., Scikit-learn: Machine learning in python, the JMLR 12 (2011) 2825–2830.

[12] A. Z. Hettinger, J. T. Cushman, M. N. Shah, K. Noyes, Emergency medical dispatch codes association with emergency department outcomes, Prehospital Emerg. Care 17 (2013) 29–37.

[13] S. A. Waien, Linking large administrative databases: a method for conducting emergency medical services cohort studies using existing data, Ac. Emerg. Med. 4 (1997) 1087–1095.

[14] J. J. Feigenbaum, Automated census record linking: A machine learning approach (2016).

[15] G. Ilangovan, Benchmarking the Effectiveness and Efficiency of Machine Learning Algorithms for Record Linkage, Master's thesis, 2019.

[16] P. Kaur, et al., A comparison of machine learning classifiers for use on historical record linkage, Master's thesis, 2020.

[17] J. M. Bronstein, C. T. Lomatsch, et al., Issues and biases in matching medicaid pregnancy episodes to vital records data: the arkansas experience, MCHJ 13 (2009) 250–259.

[18] E. Joffe, et al., A benchmark comparison of deterministic probabilistic methods for defining manual review datasets in duplicate records reconciliation, JAMIA 21 (2014) 97–104.