Class ratio and its implications for reproducibility and performance in record linkage[*]

Jeremy Foxcroft[**] [1] [0009−0002−3250−605X], Peter Christen [2] [0000−0003−3435−2015], and Luiza Antonie [1] [0000−0001−9718−3795]

[1] School of Computer Science, Reynolds Building, University of Guelph, 474 Gordon St., Guelph ON, Canada N1G 1Y4
{jfoxcrof,lantonie}@uoguelph.ca

[2] Scottish Centre for Administrative Data Research, University of Edinburgh, Bayes Centre, 47 Potterrow, Edinburgh EH8 9BT
peter.christen@ed.ac.uk

Abstract. Record linkage is the process of identifying and matching records from different datasets that refer to the same entity. This process can be framed as a pairwise binary classification problem, where a classification model predicts if a pair of records match (i.e., refer to the same entity) or not. Even though training data is paramount in model building and the subsequent predictions, there is a lack of reporting in the literature on training data details, especially the ratio of matching to non-matching examples. The absence of adequate reporting has a significant impact on both the model building and reproducibility of research studies. In this paper we demonstrate how the performance measures commonly used in record linkage (precision, recall, and F_1-measure) vary with respect to this ratio. Specifically, we show that different class imbalance ratios in training data have a substantial impact in classifier performance, with more imbalanced training data resulting in lower performance. Furthermore, we examine the impact on performance when the class ratio between the test data and the training data is changed. Our extensive experimental study allows us to offer practical advice for constructing training data, building record linkage models, measuring performance, and reporting on the training data details.

Keywords: training data · entity resolution · reproducibility · class imbalance · evaluation · precision · recall · F_1-measure

[*] We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [Discovery Grant: Bias and Representativeness in Linked Data]. This work was partially supported by the *Scottish Historic Population Platform* (SHiPP), and the support of the UK Economic and Social Research Council (ESRC) through project ES/W010321/1, is gratefully acknowledged.

[**] Corresponding author

1 Introduction

Record linkage, also known as deduplication and entity resolution, is the process of finding records or entities which refer to the same underlying entity across a single or multiple datasets [7]. Record linkage is a challenging problem particularly for datasets that are heterogeneous and contain records with poor data quality [15,17], which is the norm with most real-world applications. Considering the complexity of the record linkage problem and its associated data challenges, research continues in this field with new models being proposed and investigated across diverse research and application domains [1, 2, 5, 6, 9, 21].

The record linkage problem is often framed as a binary classification problem [20], where every pair of records in the input space is assessed and predicted to either be a match (two records referring to the same entity) or a non-match (two records referring to two different entities). The supervised machine learning models that perform this task are trained on pairs of records that have been labeled as either matching or non-matching, and it is from these labeled record pairs that the classification models are able to learn patterns and form decision boundaries to make predictions about previously unseen pairs of records. Thus, it is this training data that is the pivotal factor in model building and subsequent making of predictions. Yet there is a lack of reporting on training data details in the literature [11, 19, 23, 25], even though solutions have been proposed for data and model reporting [12, 13, 24].

This gap has significant implications for understanding the true performance of models, as well as for ensuring reproducibility and transparency. More recently, conferences and journals have started to recommend or require checklists for reproducibility [27]. This is a step in the right direction, but to our knowledge there has not been a checklist or a requirement to report class ratios. A large study on reproducibility [18] in a variety of domains where supervised classification is employed reports many articles that do not report on the training/testing split(s) used.

Training data is costly to collect, particularly in the context of record linkage. In addition, for record linkage, most of the available training data consists of positive examples and that is what is usually described in publications; the number and selection process for negative examples is not so commonly detailed. This is significant, as in record linkage it is natural for the number of negative examples to vastly exceed the number of positive examples [7].

In this paper we investigate how the training data class ratio affects the performance of classification models. Is it possible to reliably report higher performance on domain standard performance measures (such as the F_1-measure [8, 29]) by varying the number of labeled non-matching examples relative to the number of labeled matching examples during training and/or testing? Should the impact of varying these class ratios be negligible or random, this would not be an issue. If, on the other hand, varying these ratios biases the reported performance in a predictable fashion, then it is problematic to not report these numbers.

We demonstrate through our extensive experimental study that both the training and testing class ratios matter, and we investigate the effect of these ratios on different deployment scenarios. Our study allows us to offer practical advice for constructing training data, building record linkage models, and reporting on the training data details.

2 Methodology

In this section we provide a description of the methodology used to assess the impact of varying the number of matching pairs to non-matching pairs in the training and testing data on the performance of a record linkage model. Our methodology involves performing supervised record linkage on multiple benchmark datasets. Our focus is on building and measuring the performance for the classification models. We assume that cleaning, standardization, blocking and feature engineering have already been performed [7].

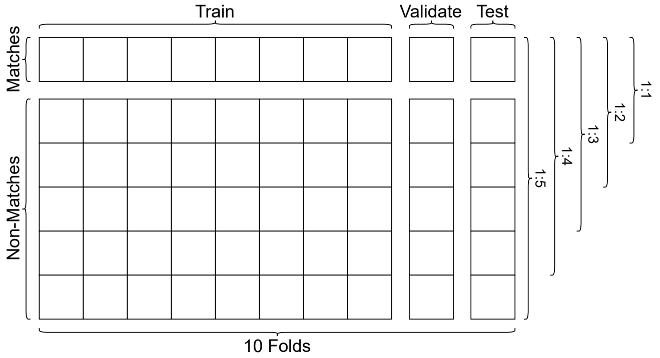
2.1 Data Partitioning

We start by separating a dataset of labeled pairs into the sets M (matches, where the two records in each pair refer to the same entity) and N (non-matches, where the records in each pair refer to different entities). We randomly sample record pairs from N such that the number of sampled pairs does not exceed $5 \times |M|$ (this is an upper bound dictated by the datasets we explore in our experimental study, shown in Table 1). From these sampled non-matching record pairs and the full set of matching record pairs M, we form 10 stratified folds for 10 fold cross validation [16]. We use a 8:1:1 ratio, where in each of the 10 runs 8 folds are used for training, 1 fold is used for validation, and 1 fold is used for testing.

The classification problem in record linkage is generally framed as a single label binary class prediction. During the training stage we build our classification model on different class ratios. We subset the non-matching examples within each fold to achieve the desired ratio of matching to non-matching pairs, considering five ratios of matching to non-matching pairs: 1:1, 1:2, 1:3, 1:4, and 1:5.

We consider the ratio of matching to non-matching pairs separately in both training and testing phases, performing a 5×5 grid search. In most research applications, models are evaluated on a test set that contains the same class imbalance as the training data. In more practical situations where record linkage models are deployed on previously unseen real-world data, it is often not possible to know the class ratio of the deployment class while the model is being trained. As such, it is also interesting to look at the impact that a relative abundance or deficit of non-matching pairs during training has on model performance when deployed in a different scenario than encountered in the training phase. The partitioning of the data across the ten folds and five ratios is shown in Figure 1.

Fig. 1. Match and non-match allocation across folds and ratios, as described in section 2.1



2.2 Classification and Evaluation

To perform the pairwise classification, we train models on the training folds described in Section 2.1. Given a pair of records, a model returns the probability that they refer to the same entity. During this stage, we evaluate the performance of our classification model on the test set. The performance is measured through precision (the fraction of all positive predictions that are actual positives), recall (the fraction of all actual positives that are predicted to be positive) and F_1-measure (the harmonic mean of precision and recall). Expanding these definitions, F_1-measure can be defined in terms of the number of true positives (TP), false positives (FP), and false negatives (FN) as $F_1 = 2 \times TP/(2 \times TP + FP + FN)$ [8].

To calculate F_1-measure, the number of TPs, FPs, and FNs within the test set must first be identified. A classification model returns the conditional probability p that an individual pair is a match. A decision rule in the form of a numeric threshold $t \in [0, 1]$ is required to convert these probabilities into binary predictions (i.e., only pairs where $p > t$ are considered to match). Once a pair has been assigned a binary prediction, it can be identified as a TP, FP, or FN.

A threshold $t = 0.5$ may seem a natural starting point, but using a threshold other than 0.5 to assign binary predictions can often increase the performance as measured by the F_1-measure [22]. We select a threshold for each model that maximizes the F_1-measure on the validation set, and use this threshold when computing precision, recall, and the F_1-measure for the test set.

3 Experimental Study

We now introduce the datasets we use in our experimental study and present our results. All results and source code to reproduce these experiments are available in a public GitHub repository.[3] In this repository, we also report results on additional performance measures.

[3] https://github.com/foxcroftjn/PAKDD-Class-Ratio

Table 1. The datasets, labeled matches, labeled non-matches, and (rounded) class ratios taken from [28].

Dataset Name	Matches	Non-Matches	Ratio
abt-buy	1 095	6 067	1:5
amazon-google	1 298	7 142	1:5
walmart-amazon	1 154	14 425	1:12
wdc_xlarge_computers	9 991	59 571	1:5
wdc_xlarge_shoes	4 440	39 088	1:8
wdc_xlarge_watches	9 564	53 105	1:5

3.1 Datasets

In our study, we use a variety of record linkage datasets that were prepared by Primpeli and Bizer [28] and available online.[4] As part of their work, they published labeled pairs of records and similarity vectors suitable for supervised machine learning models (e.g., Random Forests [3], Support Vector Machines [10] classifiers) for 21 publicly available labeled datasets for record linkage.

We report results only for the datasets with at least 1 000 labeled matching pairs. This filters out a number of the less comprehensively labeled datasets. In addition, we require each dataset to contain at least five labeled non-matching pairs for every matching pair, to ensure we can train and test a model on a 1:5 class imbalance without compromising the set of record pairs which characterize the matching class. Finally, we do not use datasets where the Random Forests model in [28] achieved an F_1-measure ≥ 0.99, as these linkage tasks are too easy [26] to draw meaningful conclusions from in our work. After applying this filtering criteria we are left with the six datasets summarized in Table 1.

3.2 Results

We consider three different architectures of classification model: Random Forest (RF) [3], Support Vector Machine (SVM) [10], and Entity Matching Transformer (EMT) [4]. The first two model architectures are traditional machine learning techniques; the third uses the roBERTa attention-based transformer architecture to achieve near state-of-the-art results. We rely on existing implementations for each of these architectures [4, 28]; the specific details of how the models are implemented is outside the scope of this work. By performing this experiment using both traditional machine learning and deep learning, we demonstrate that our results are not an artifact of a specific classification method, but rather that they generalize across a variety of commonly used classification approaches.

Figure 2 shows the F_1-measure results for all the architecture/dataset combinations. It is interesting to observe that a consistent gradient has emerged. The left column for each architecture/dataset, where the test set contains the

[4] https://data.dws.informatik.uni-mannheim.de/benchmarkmatchingtasks

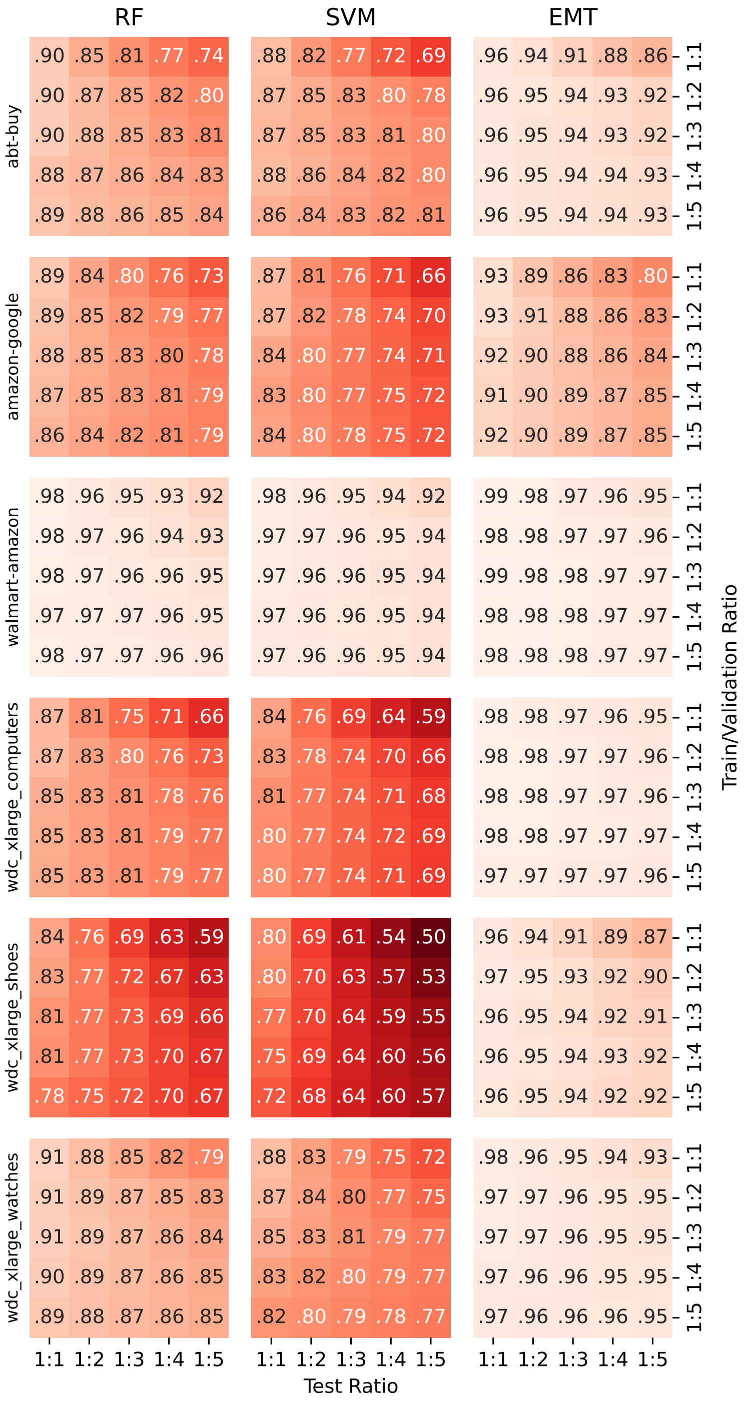


Fig. 2. F_1-measure for all model architectures, train/test ratio combinations, and datasets.

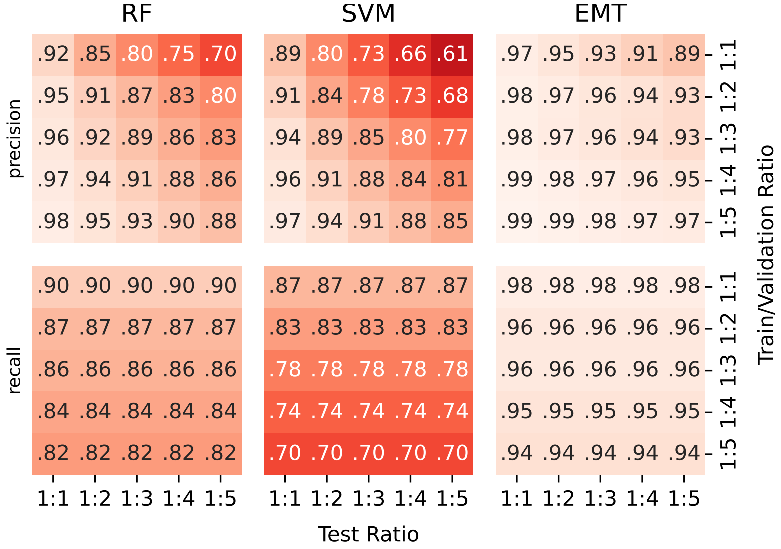


Fig. 3. Precision and recall for all model architectures on the amazon-google dataset.

fewest non-matching pairs, consistently contains the highest reported performance. Moving from the top left to the bottom right consistently causes F_1-measure performance to monotonically decrease. This tells us that we can worsen the reported performance of a methodology just by increasing the number of non-matching pairs in the training and testing data (or conversely, we can bolster the reported performance by reducing the number of non-matching pairs). Most work published in record linkage lies somewhere on this diagonal, as it is normal to train and test on data that have a single fixed class ratio in a controlled environment. The problematic part is that when class ratios are not reported, it is unknown where on this diagonal reported results lie. Class ratios higher than 1:5 are regularly used when performing record linkage [28], which in turn leads to the reported F_1-measure results skewing even lower than in this experiment.

Another commonality across all the architecture/dataset combinations is that the top right corner consistently contained the lowest reported F_1-measure performance, whereas the performance in the bottom left was in some cases the highest we obtained in our evaluation. We believe this trend can be used to inform the training data selection for real-world models, where the class ratio of the deployment data is not known when training a model. The low number in the top right corner reflects the consequences of training with a lower class ratio than testing. In contrast, the bottom left number reflects the consequences of a relative abundance of non-matching pairs during training. As such, we conclude that it is better to overestimate than underestimate the class imbalance of the deployment environment when choosing the class imbalance of a training set.

When looking at precision, we observe that it monotonically decreases when we use a fixed training ratio and gradually increase the non-matching pair count in the test set. It is helpful to remember that each 1:n test set is a superset of

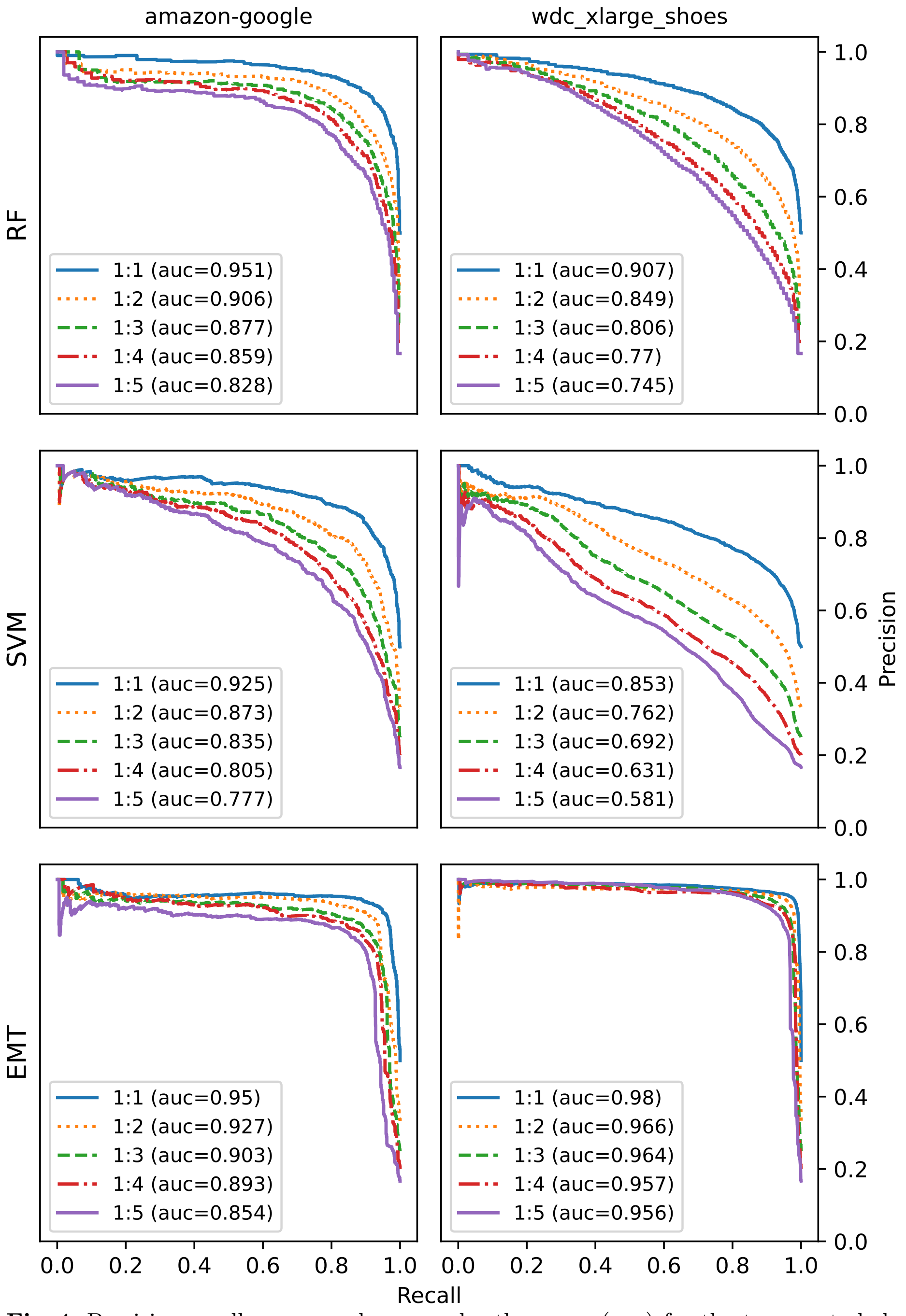


Fig. 4. Precision-recall curves and area under the curve (auc) for the two most challenging datasets. Only experiments with the same train/test ratio are shown.

Table 2. The binary classification thresholds for experiments with the same train/test ratio. Thresholds reported as 0.00 are less than 0.005 (but still greater than 0).

Dataset Name	RF					SVM					EMT				
	1:1	1:2	1:3	1:4	1:5	1:1	1:2	1:3	1:4	1:5	1:1	1:2	1:3	1:4	1:5
abt-buy	0.45	0.43	0.36	0.44	0.39	0.41	0.48	0.45	0.34	0.40	0.00	0.00	0.05	0.01	0.00
amazon-google	0.49	0.45	0.46	0.42	0.43	0.46	0.37	0.35	0.36	0.28	0.00	0.00	0.00	0.00	0.00
walmart-amazon	0.44	0.40	0.45	0.48	0.46	0.46	0.53	0.45	0.47	0.43	0.00	0.99	0.06	0.96	0.07
wdc_xlarge_computers	0.44	0.42	0.44	0.40	0.38	0.41	0.39	0.37	0.30	0.23	0.02	0.02	0.01	0.10	0.04
wdc_xlarge_shoes	0.43	0.39	0.38	0.34	0.38	0.35	0.26	0.24	0.20	0.17	0.00	0.00	0.00	0.02	0.01
wdc_xlarge_watches	0.50	0.48	0.43	0.44	0.43	0.49	0.39	0.37	0.34	0.33	0.97	0.79	0.05	0.33	0.31

the 1:$(n-1)$ test set, differing only through the addition of more non-matching pairs. Since precision is $TP/(TP+FP)$, we can justify this monotonic decrease by observing that an increase in negatives in the test set will not affect the TP count, but will increase the FP count (unless all new non-matching pairs in the test set are correctly labeled by the classifier).

When looking at recall, we observe that performance for a fixed training ratio is invariant. Since recall is defined as $TP/(TP+FN)$, this consistency can be explained by remembering that increasing the number of non-matches in the test set affects neither the TP or the FN count. It is important to remember that precision and recall are not equally important for all applications, and that either of these metrics can always be increased at the expense of the other [14,22]. Precision and recall for the three architectures and the amazon-google dataset are shown in Figure 3.

Our initial approach to measuring record linkage performance was to use a threshold of 0.5 on the classification function C, which yielded results with the same gradients as those shown in Figure 2. To investigate whether the gradients were an artifact of this likely sub-optimal threshold, we instead calculate thresholds as discussed in Section 2.2. These thresholds, shown in Table 2, were used to calculate the F_1-measure values in Figure 2. The results when using a fixed threshold of 0.5 are available in the GitHub repository mentioned at the beginning of Section 3.

Even with this updated approach to computing F_1-measure, representing model performance using a single number does not make for a comprehensive comparison. We address this by showing precision-recall curves in Figure 4. We only show curves from the diagonals in Figure 2 where the training and testing ratios are the same, as this is the context where most published results lie. We also choose to focus on the two most challenging datasets, although our analysis is consistent with the curves that are not shown. From the precision-recall curves, it is visible that model performance suffers at almost all thresholds by increasing the ratio of matches to non-matches in the training and testing data, not just at the 0.5 threshold or the threshold which seeks to maximize F_1-measure.

4 Discussion and Recommendations

The two key findings of our work are:

1. The F_1-measure can be artificially lowered or raised in a predictable direction by increasing the number of non-matching pairs in the training and/or testing sets of a record linkage problem. The impact of varying this ratio means that it is important to report this ratio when reporting the results of a record linkage methodology. Stating the number of labeled matching pairs is not sufficient.
2. When the deployment environment class imbalance is unknown, it is safest to err on the side of including more non-matching pairs during training. There is a consistently larger performance penalty to underestimating this ratio as compared to overestimating it.

Following from these findings, our recommendations for building training data and reporting are as follows:

1. Document [12] and report on the construction of the training data.
2. Report the class ratio or both the number of matching and non-matching pairs that are used to build the classification model used in record linkage.
3. Add more non-matching pairs to the training data when the class imbalance in the deployment environment is unknown.

Finally, it is worth commenting on the training time required for each of the model architectures. Each Random Forest took only seconds to train, as the training process parallelizes effectively across a multi-core CPU. Training the SVM models does not naturally parallelize, so models sometimes took a couple minutes to train (also using only a CPU). The EMT models were trained for 3 hours each using a GPU. It is worth considering if the higher performance of a deep learning approach is always worth the significantly higher demand for specialized hardware and training time.

5 Conclusions and Future Work

There are many factors that come into play when preparing training data for binary classification of record linkage problems. The focus of this paper is on how class ratio affects F_1-measure, one of the most common performance measures used for classification and record linkage problems.

The impact of other aspects of training data creation remain the potential subject of future work. This includes how labeled non-matching pairs are sourced (i.e., random sampling? hard negative mining [11, 28]?) and strategies for reducing the size of the training data to accelerate model training without compromising model quality. Other future research directions are to investigate if the findings discovered in this study hold for multi class settings and other application domains where imbalanced classes are common.

References

1. Akgün, Ö., Dearle, A., Kirby, G.N.C., Christen, P.: Using metric space indexing for complete and efficient record linkage. In: Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD. pp. 89–101. Springer, Melbourne (2018). https://doi.org/10.1007/978-3-319-93040-4_8
2. Anindya, I.C., Kantarcioglu, M., Malin, B.: Determining the impact of missing values on blocking in record linkage. In: Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD. p. 262–274. Springer, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-030-16142-2_21
3. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (Oct 2001). https://doi.org/10.1023/A:1010933404324
4. Brunner, U., Stockinger, K.: Entity matching with transformer architectures - a step forward in data integration. Proceedings of the 23rd EDBT (Mar 2020). https://doi.org/10.21256/ZHAW-19637
5. Cao, X., Zheng, Y., Shi, C., Li, J., Wu, B.: Link prediction in schema-rich heterogeneous information network. In: Advances in Knowledge Discovery and Data Mining: 20th Pacific-Asia Conference, PAKDD. pp. 449–460. Springer, Auckland (2016)
6. Cao, Y., Peng, H., Yu, P.S.: Multi-information source hin for medical concept embedding. In: Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD. p. 396–408. Springer, Berlin, Heidelberg (2020). https://doi.org/10.1007/978-3-030-47436-2_30
7. Christen, P.: Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Data-Centric Systems and Applications, Springer (2012)
8. Christen, P., Hand, D.J., Kirielle, N.: A review of the F-measure: Its history, properties, criticism, and alternatives. ACM Comput. Surv. **56**(3) (2023)
9. Christen, P., Ranbaduge, T., Schnell, R.: Linking Sensitive Data. Springer, Heidelberg (2020)
10. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273–297 (Sep 1995). https://doi.org/10.1007/BF00994018
11. Fakhraei, S., Mathew, J., Ambite, J.L.: NSEEN: Neural semantic embedding for entity normalization. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, 2019. p. 665–680. Springer-Verlag, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-030-46147-8_40
12. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H., III, H.D., Crawford, K.: Datasheets for datasets. Commun. ACM **64**(12), 86–92 (Nov 2021). https://doi.org/10.1145/3458723
13. Gilbert, R., Lafferty, R., Hagger-Johnson, G., Harron, K.L., Zhang, L.C., Smith, P., Dibben, C., Goldstein, H.: Guild: Guidance for information about linking data sets†. Journal of Public Health (Oxford, England) **40**, 191–198 (2017)
14. Hand, D.J., Christen, P.: A note on using the F-measure for evaluating record linkage algorithms. Statistics and Computing **28**(3), 539–547 (2018)
15. Harron, K., Dibben, C., Boyd, J., Hjern, A., Azimaee, M., Barreto, M.L., Goldstein, H.: Challenges in administrative data linkage for research. Big Data & Society **4**(2) (2017). https://doi.org/10.1177/2053951717745678, pMID: 30381794
16. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA (2001)

17. Herzog, T., Scheuren, F., Winkler, W.: Data Quality and Record Linkage. Springer (01 2007). https://doi.org/10.1007/0-387-69505-2
18. Kapoor, S., Narayanan, A.: Leakage and the reproducibility crisis in ML-based science (2022). https://doi.org/10.48550/ARXIV.2207.07048
19. Kooli, N., Allesiardo, R., Pigneul, E.: Deep learning based approach for entity resolution in databases. In: Nguyen, N.T., Hoang, D.H., Hong, T., Pham, H., Trawinski, B. (eds.) Intelligent Information and Database Systems - 10th Asian Conference, ACIIDS 2018, Dong Hoi City, 2018. pp. 3–12. Springer (2018). https://doi.org/10.1007/978-3-319-75420-8_1
20. Köpcke, H., Rahm, E.: Frameworks for entity matching: A comparison. Data and Knowledge Engineering **69**(2), 197–210 (2010). https://doi.org/https://doi.org/10.1016/j.datak.2009.10.003
21. Koumarelas, l., Papenbrock, T., Naumann, F.: Mdedup: Duplicate detection with matching dependencies. Proc. VLDB Endow. **13**(5), 712–725 (Jan 2020). https://doi.org/10.14778/3377369.3377379
22. Lipton, Z.C., Elkan, C., Naryanaswamy, B.: Optimal thresholding of classifiers to maximize F1 measure. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 225–239. Springer, Berlin, Heidelberg (2014)
23. Makri, C., Karakasidis, A., Pitoura, E.: Towards a more accurate and fair SVM-based record linkage. In: Tsumoto, S., Ohsawa, Y., Chen, L., den Poel, D.V., Hu, X., Motomura, Y., Takagi, T., Wu, L., Xie, Y., Abe, A., Raghavan, V. (eds.) International Conference on Big Data. pp. 4691–4699. IEEE, Osaka (2022). https://doi.org/10.1109/BigData55660.2022.10020514
24. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., Gebru, T.: Model cards for model reporting. In: Proceedings of the Conference on Fairness, Accountability, and Transparency. p. 220–229. FAT∗'19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3287560.3287596
25. Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., Raghavendra, V.: Deep learning for entity matching: A design space exploration. In: Proceedings of the 2018 International Conference on Management of Data. p. 19–34. SIGMOD '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3183713.3196926
26. Papadakis, G., Kirielle, N., Christen, P., Palpanas, T.: A critical re-evaluation of benchmark datasets for (deep) learning-based matching algorithms. In: IEEE International Conference on Data Engineering (ICDE). Utrecht (2024)
27. Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché Buc, F., Fox, E., Larochelle, H.: Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). J. Mach. Learn. Res. **22**(1) (Jan 2021)
28. Primpeli, A., Bizer, C.: Profiling entity matching benchmark tasks. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. pp. 3101–3108. CIKM '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3340531.3412781
29. Shaw, W., Burgin, R., Howell, P.: Performance standards and evaluations in IR test collections: Cluster-based retrieval models. Information Processing & Management **33**(1), 1–14 (1997). https://doi.org/10.1016/S0306-4573(96)00043-X