Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, pp. 7-12, Washington DC, August, 2003

On Evaluation and Training-Set Construction for Duplicate Detection

Mikhail Bilenko and Raymond J. Mooney
Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712
{mbilenko,mooney}@cs.utexas.edu

ABSTRACT

A variety of experimental methodologies have been used to evaluate the accuracy of duplicate-detection systems. We advocate presenting precision-recall curves as the most informative evaluation methodology. We also discuss a number of issues that arise when evaluating and assembling training data for adaptive systems that use machine learning to tune themselves to specific applications. We consider several different application scenarios and experimentally examine the effectiveness of alternative methods of collecting training data under each scenario. We propose two new approaches to collecting training data called static-active learning and weakly-labeled non-duplicates, and present experimental results on their effectiveness.

1. INTRODUCTION

Properly evaluating the accuracy of duplicate detection requires a "gold-standard" dataset in which *all* duplicate records have been identified. A gold-standard dataset therefore consists of a set of equivalence classes of records, where an equivalence class contains all the records in the database referring to a particular entity. Measuring the ability of an algorithm to correctly identify equivalence classes in gold-standard annotated data is the best approach to assessing its accuracy. Since the relative costs of labeling a non-duplicate as a duplicate (false positives) and overlooking true duplicates (false negatives) can vary across applications, we believe the best experimental results to present are precision-recall (PR) curves. Few previously published evaluations of duplicate-detection have presented PR curves, therefore we propose a recommended methodology for generating PR curves based on standard practices in information retrieval.

Several other methodological issues arise when training adaptive duplicate-detection systems using machine learning. These include how to efficiently collect effective training data for the system and how to appropriately measure generalization accuracy. We can imagine two different scenarios in which machine learning can be used to improve duplicate detection. In the first scenario, the goal is to use machine learning to develop a general duplicate-detection system tailored to a specific *type* of data, such as mailing addresses or bibliographic citations, but not tailored to a specific database. In this approach, the eventual databases to be cleaned are not available during the training phrase. We call this the "shrink-wrap" scenario, since the goal is to develop and market a static "shrink-wrapped" software system that any user can apply to their own database without further training. In the second scenario, the goal is to train a system to clean a specific database, and sample duplicate and non-duplicate pairs from this database can be identified by the user during the training phase. We call this the "consulting" scenario, since it seems most appropriate under a business model where a company is hired to clean specific databases and trains the software specifically for each database.

In both scenarios, results must be gathered on disjoint test data to properly determine accuracy. However, one would ideally also like to measure how rapidly system performance decreases as the distribution of the training data varies from examples from the same database (consulting scenario) to examples from other databases of decreasing similarity (shrink-wrap scenario). We present an experimental methodology and sample results examining these issues using several extant gold-standard datasets.

A distinct but related issue is how labeled training examples are obtained. One approach is to require the user to identify all duplicates in a random sample of records or in the entire database. Note that for N records, this requires $O(N^2)$ comparisons in the worst case. Another approach is to randomly choose pairs of records and ask the user to label them as "same" or "different". Finally, various strategies may be used to actively select "good" training pairs from the available data. Since most pairs of records selected at random will *not* be equivalent, selecting only pairs that are fairly similar accordingly to some default, static metric, may be a good strategy. More sophisticated active learning strategies that dynamically select the next best pair of examples based on the current results of learning can be very useful [16, 17]. We present results on various simple static sample-selection strategies and make several methodological suggestions based on the results.

Finally, we present an unsupervised strategy for obtaining negative training examples. Since in a typical database the vast majority of randomly selected record pairs are non-duplicates, it is possible to populate the training set with negative examples based on such pairs, while filtering out likely pairs of duplicate records using off-the-shelf similarity metrics such as vector-space cosine similarity. We present experimental results that prove the viability of such a strategy for obtaining "weakly-labeled" negative examples without user supervision.

ACM SIGKDD-03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation

2. BACKGROUND

The problem of identifying database records that are syntactically different yet describe the same physical entity has been referred to as duplicate detection [13], identity uncertainty [14], object identification [18], and deduplication [16]. Record linkage is a variation of the problem that arises when records that describe the same entity are matched across multiple databases [6, 19]. Duplicate detection systems go through the process of identifying matching pairs via the following three phases:

1. Candidate generation. Since evaluating all possible record pairs is highly inefficient because most of them are clearly dissimilar non-matches, only record pairs that are loosely similar (e.g. share common tokens) are selected as candidates for matching using "blocking" [9] or "canopies" [12] techniques.

2. Similarity calculation. For every candidate pair of records (R_i, R_j) identified in step 1, similarity $Sim(R_i, R_j)$ is computed using some distance metric(s) or a probabilistic method.

3. Linkage and closure. Candidate pairs that have similarity scores higher than a threshold value T_{sim} are linked; transitive closure of those linked points forms the final equivalence classes of duplicate records.

Adaptive duplicate detection systems [4, 5, 16, 18, 20] attempt to improve the accuracy of matching by exploiting labeled training data in the form of record pairs that are marked as duplicates or non-duplicates by the user. While methods that improve computational efficiency of record linkage have received some attention [1, 9], the primary focus of research on machine learning methods for deduplication has been on improving accuracy.

In this paper, we conduct experiments using the MARLIN duplicate detection system [4] which uses labeled training examples at two levels. First, MARLIN can utilize trainable string metrics, such as learnable edit distance, that adapt textual similarity computations to specific record fields. Second, MARLIN trains a classifier to discriminate between pairs of duplicate and non-duplicate records using textual similarity values for different fields as features.

Three datasets are used in the experiments: *Cora* is a collection of 1295 distinct citations to 122 research papers, where each citation has been segmented into fields such as **author**, **title**, **venue**, etc.. The *Citeseer* dataset is comprised of 1564 single-field records that represent citations to 785 unique papers. Finally, *Restaurant* is a database containing 864 restaurant records that contain 112 duplicates. Each record is composed of four fields: **name**, **address**, **city** and **cuisine**.

All experiments are conducted using two-fold cross-validation. Folds are created by randomly assigning equivalence classes of duplicate records, except for experiments in Section 4.1, where pairs of duplicate records are randomly assigned to folds. All results are reported over 20 runs, where for each run the two folds are used alternately for training and testing.

3. EVALUATION MEASURES FOR DUPLICATE DETECTION

There has been little work in comparing the accuracy of various adaptive deduping techniques experimentally, a problem that is partially caused by scarcity of publicly available benchmark datasets. Accuracy comparisons between different systems are also hindered by the fact that a variety of different measures have been used to evaluate individual approaches, such as:

- Maximum F-measure, which is the harmonic mean between pairwise precision and recall [4, 5, 16];
- Pairwise precision for the optimal number of pairs [18];
- Percentage of the correct equivalence classes for which an error exists in the grouping [11, 14];
- Proportions of true matching pairs at fixed error levels [20].

Although all of these quantities characterize accuracy of duplicate detection systems, they sidestep the problem of selecting the threshold T_{sim} that separates duplicates from non-duplicates. These single-value accuracy measures either assume the optimal value of T_{sim} has been chosen by the user, or select a certain value implicitly, as maximum F-measure does. One problem with this approach is that the relative cost of false positives (non-duplicate pairs selected as duplicates) and false negatives (unidentified duplicate pairs) may vary, making the optimal value of T_{sim} situation-specific . Additionally, the record-linkage literature has advocated using two thresholds, T_{auto} and T_{manual}, separating pairs of records into three classes: those that are not linked ($Sim < T_{manual}$), those that should undergo human review ($T_{manual} \leq Sim \leq T_{auto}$), and those that should be linked ($Sim > T_{auto}$) [6, 19].

Precision-recall curves, traditionally used for evaluating information retrieval systems [2], provide a method for presenting performance over the complete range of possible threshold values. Precision and recall for duplicate detection are calculated based on the number of duplicate pairs found by the system: precision is the proportion of identified duplicate pairs that are correct, and recall is the proportion of actual duplicate pairs in the test database that have been identified. Precision values are interpolated at 20 standard recall levels following the traditional procedure in information retrieval [2]. The curves are obtained by successive lowering of the threshold value, labeling pairs whose similarity is above the threshold as duplicates, and updating the transitive closure to obtain the equivalence classes of identified duplicates at each distinct point.

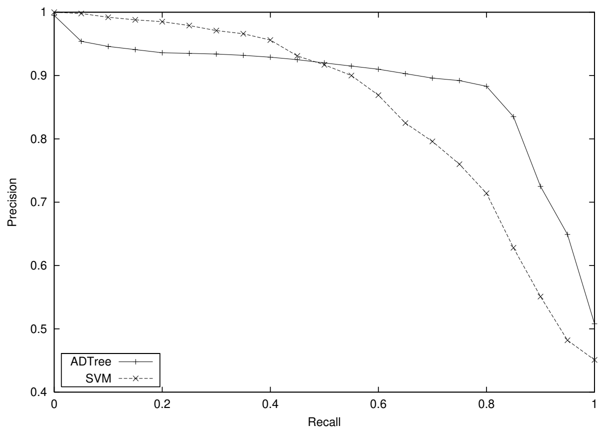

Figure 1: Deduplication precision of two classifiers for different recall levels on the *Cora* dataset.

Figure 1 presents results of an experiment where precision-recall curves illustrate behavior that would be overlooked by single-value comparisons. It compares two adaptive record linkage systems that use different state-of-the-art classifiers to discriminate between duplicate and non-duplicate pairs: SVM^{light} [10] and alternating decision trees [7]. Both systems employ edit distance with affine gaps [8] as the underlying string comparison metric. System performance is compared on the *Cora* dataset; training is performed using

40 randomly selected pairs of duplicate records and 40 randomly selected pairs of non-duplicate records.[1] While the ADTree-based system outperforms the SVM-based system for high recall values, it makes more mistakes at low recall, and therefore is inferior in situations when a small set of highly accurate duplicate pairs is desired.

When adaptive systems are being compared, learning curve plots illustrate the dependence of accuracy on the amount of training data, which is provided as record pairs that are labeled as duplicates or non-duplicates by a human expert. Learning curves are particularly important for evaluating the utility of active learning techniques that attempt to select record pairs that, when labeled, are most useful for improving the matching accuracy [16, 18]. Figure 2 demonstrates how precision-recall curves can be combined with learning curves on three-dimensional plots, providing a comprehensive comparison of adaptive deduplication systems.

As in Figure 1, the SVM-based and ADTree-based systems described above are compared on the *Cora* dataset. Equal amounts of randomly selected duplicate and non-duplicate record pairs were used to comprise the training set at every point on the "Training Pairs" axis. Performance of each system is described by a surface. Its projections onto the precision-recall plane yield precision-recall curves for fixed amounts of training data, while projections onto the precision-training plane produce learning curves for fixed recall levels. The surface plot for one system lying higher than for another indicates that the system performs better at a specific recall level for a given amount of training data. In Figure 2, the ADTree-based system outperforms the SVM-based system at high recall levels for all amounts of training data, as well as at at low recall levels when a moderate amount of training data is supplied (15-30 training examples). Thus, the optimal classifier choice for duplicate detection on this dataset depends both on the desired precision/recall and on the amount of available training data.

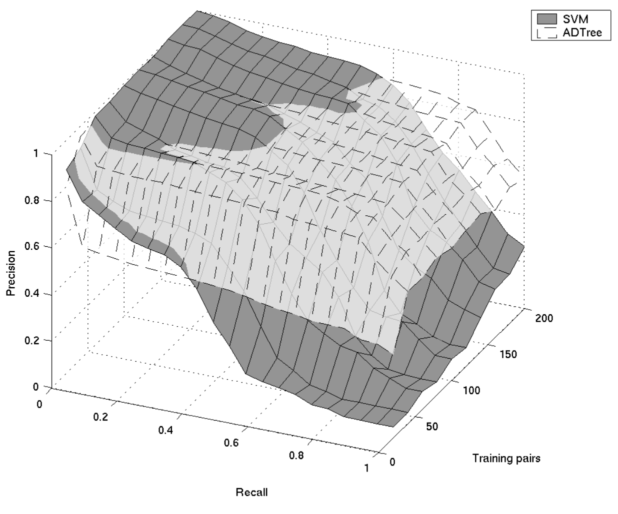

Figure 2: Comparison of two adaptive systems for varying amounts of training data on the *Cora* dataset.

Overall, we argue that precision-recall curves allow comprehensive comparisons of duplicate detection systems for varying relative misclassification costs and for different amounts of training data, and therefore should be used for experimental evaluation of deduplication systems over single-value accuracy measures. PR curves can also be used for selecting the optimal similarity threshold T_{sim} empirically and to aid and validate statistical approaches to record linkage error rate estimation [3]. The importance of using PR curves for evaluation of duplicate detection systems echoes the findings of Provost *et. al.* [15], who advocate using ROC curves for comparing classifiers over single-value accuracy measures.

[1] In these experiments we assume for simplicity that training data was provided by a human expert. Section 4 discusses realistic scenarios for training set construction in detail.

4. TRAINING SCENARIOS FOR ADAPTIVE DEDUPLICATION SYSTEMS

4.1 "Consulting" versus "Shrink-wrap"

Circumstances in which duplicate detection systems are used in an industrial setting can vary significantly. Ideal conditions for training an adaptive deduplication system include availability of a fair-sized subset of the database with all equivalence classes of duplicates identified. A training set of same-class and different-class record pairs can then be created from such a subset with labeled equivalence classes. We refer to such a scenario where training data is extracted from the specific database to be deduped as the *consulting* framework, since it emulates the process of on-site system deployment by a consultant.

Sometimes no identified duplicate or non-duplicate record pairs are available from the actual database for privacy or cost-saving reasons. We refer to such cases when no training data from the test database is available as the *shrink-wrap* scenario: the deduplication system must be tuned using labeled records from other databases whose similarity to the database being deduped may vary.

An ideal shrink-wrap system is specialized to a certain type of data (e.g. census records) and trained on a dataset that is similar to the actual database. We have conducted experiments that utilize trainable edit distance, an adaptable string similarity metric, to investigate the performance of a duplicate detection system on the continuum from the consulting scenario to shrink-wrap scenarios with various degrees of similarity between the training and testing databases. Figure 3 contains experimental results, where deduplication was evaluated on the *Citeseer* database. Trainable edit distance with affine gaps [4] was utilized for similarity comparisons between single-field citation records; training was performed using 20 randomly selected pairs of duplicate records. To simulate the the consulting scenario, record pairs where assigned to folds randomly, possibly splitting equivalence classes between the testing and training sets. Five possible training scenarios are examined:

- A consulting scenario, where edit distance is trained using randomly selected pairs from the *Citeseer* database;
- A "similar" shrink-wrap scenario, where edit distance is trained using pairs of duplicate records from the *Cora* database that also contains citations to computer science literature;
- A "dissimilar" shrink-wrap scenario, where edit distance is trained using pairs of duplicate records from the *Restaurant* database containing single-field records of restaurant names and addresses;
- A "grossly dissimilar" shrink-wrap scenario, where edit distance is trained using pairs of duplicate records from the *Restaurant* database that include only restaurant names;
- A completely unlearned scenario, where generic edit distance with affine gaps [8] is used to identify duplicates.

Results on Figure 3 show that the utility of training an adaptive string distance metric depends on the similarity between the training and testing databases. The consulting approach results in the highest accuracy, since near-optimal values of the edit-distance parameters are obtained from training on a subset of the actual testing database. When training is performed on a similar yet different database, *Cora*, performance degrades only slightly. Training the edit distance on databases that are not related to the testing dataset results in significant drops in performance that are worse than the generic unlearned edit distance. Accuracy degradation is correlated with the similarity of data between datasets: most records in the *Restaurant-Name* database are very short strings, while records in the full *Restaurant* database are longer, and therefore more similar to the *Citeseer* citation strings; therefore the system trained on *Restaurant* pairs does not do as poorly as the system trained on *Restaurant-Name* pairs.

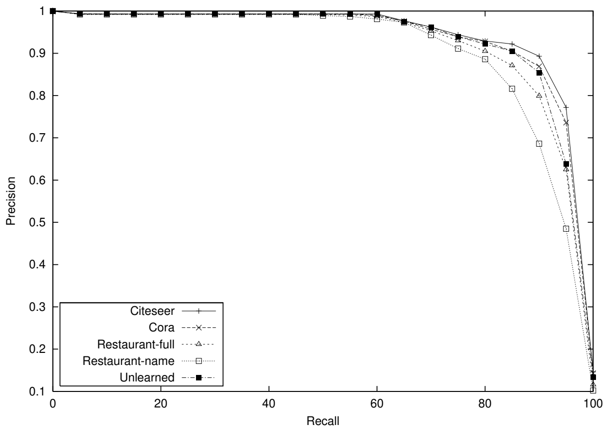

Figure 3: Comparison of consulting and shrink-wrap training scenarios for adaptive record linkage. The system was trained on the specified datasets and tested on the *Citeseer* dataset.

These results indicate that the extra effort of tuning a string metric used by the duplicate detection system on training data selected from the actual database leads to higher accuracy. However, if a system that was trained on very similar data is available, obtaining training data from the current database leads to minor improvements, and is probably worthwhile only if misclassification costs are high.

4.2 Static-active selection of duplicate records

Training adaptive duplicate detection systems in real-world scenarios involves selecting a set of record pairs for a human expert to label as duplicates or non-duplicates. Since typical databases contain small amounts of duplicate records, selecting random pairs of database records as potential training examples leads to database subsets with extremely few identified duplicates (positive examples). As a result, such randomly selected training sets are highly skewed toward non-duplicates, which leads to suboptimal performance of classifiers trained on that data.

Active learning methods attempt to identify informative examples that lead to maximal accuracy improvements when added to the training set. During each round of active learning, the example(s) that is estimated to improve performance the most when added to the training set is identified and labeled. The system is then re-trained on the training set including the newly added labeled example. Thus, traditional active learning systems are "dynamic": labels of training examples selected in earlier rounds influence which unlabeled examples are deemed most informative in subsequent rounds.

While prior work has examined active learning approaches to adaptive record linkage [16, 18], such strategies may not always be feasible due to high computational costs or logistic issues. We propose using a simple "static" active learning method for selecting pairs of records that are likely duplicates as a middle ground between computationally expensive dynamic active learning methods that try to identify the most informative training examples and random selection that is efficient but fails to select useful training data.

Our approach relies on the fact that off-the-shelf string similarity metrics, such as the Jaro metric [19] or TF-IDF vector-space distance [2], can accurately identify duplicate pairs at low recall levels even for databases where duplicates are difficult to separate from non-duplicates at high recall levels. Therefore, when a random sample of records from a database is taken and similarity between them is computed using such an off-the-shelf similarity metric, record pairs that have high similarity scores across multiple fields are likely potential examples of duplicate pairs. By asking the user to label records that have high textual similarity, a training sample with a high proportion of duplicates can be obtained. At the same time, non-duplicate records selected using this method are likely to be "near-miss" negative examples that are more informative for training than randomly selected record pairs most of which tend to be "easy" non-duplicates.

Figures 4 and 5 demonstrate the comparative utility of static-active selection and random pair selection for choosing training record pairs on *Restaurant* and *Cora* datasets respectively. Each system was trained on 40 training examples comprised of randomly selected record pairs and/or the most similar pairs selected by a static-active method using TF-IDF cosine similarity. Using a token-based inverted index for the vector-space model [2] allowed efficient selection of static-active training examples without computing similarity between all pairs of records. All experiments utilized SVM^{light} as the classifier and two textual similarity metrics for field comparisons: TF-IDF cosine similarity and unlearned edit distance with affine gaps.

For both datasets, the highest performance is achieved when training data is a mix of examples selected using the static-active strategy and randomly chosen record pairs. However, employing many random pairs with a few static-active examples yields the best results on *Cora*, while on *Restaurant* the highest performance is achieved when the system is trained on a balanced mix of static-active and random examples. This difference is explained by the makeup of the two datasets. *Cora* has a higher absolute number of duplicates than *Restaurant* (8592 versus 56 for each fold); duplicates in *Cora* also represent a larger proportion of all record pairs (8592/211242 versus 56/93406 for each fold). On *Restaurant*, random selection results in datasets that contain almost no duplicates, while including a significant number of pairs selected using the static-active technique leads to balanced training sets that contain sufficient positive and negative examples. On *Cora*, however, randomly selected pairs are likely to contain a few duplicates. Including a limited number of record pairs chosen using the static-active technique results in the best performance, but as more and more static-active examples are added, performance goes down because highly similar duplicates take the place of informative non-duplicates in the training set. Thus, the worst performance on *Restaurant* occurs when all training examples are chosen randomly because duplicates are almost never encountered, while on *Cora* using only examples chosen by static-active selection results in the opposite problem: extremely few non-duplicate pairs are found, and the class distribution of training data is highly skewed toward duplicates.

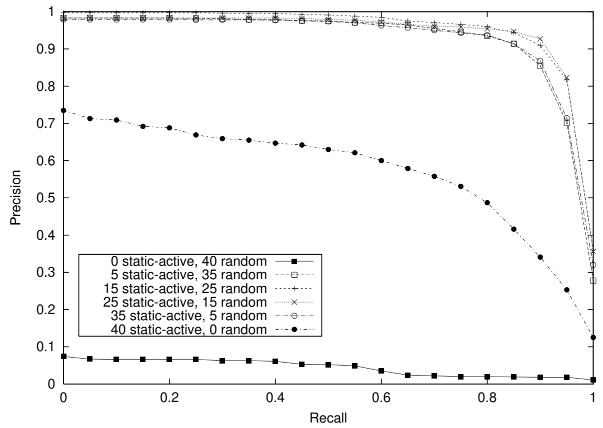

Figure 4: Comparison of random and static-active training example selection on the *Restaurant* dataset.

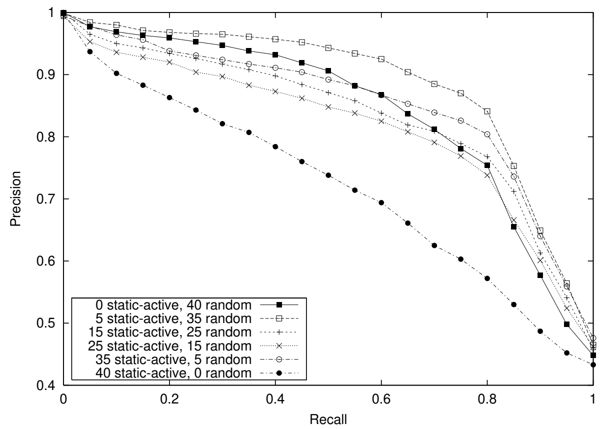

Figure 5: Comparison of random and static-active training example selection on the *Cora* dataset.

Based on these results, we conclude that best training sets for adaptive duplicate detection systems are obtained when randomly chosen pairs of records are combined with pairs chosen using static-active selection. The specific proportion in which the two kinds of training data should be mixed can be estimated based on the outcome of labeling randomly chosen pairs. If duplicates are exceptionally rare, a significant number of static-active examples is required to obtain a sufficient sample of duplicates, while databases with a large number of duplicates need only a small number of record pairs selected using the static-active methodology to complete a representative training set.

Overall, we argue that a reasonable baseline to which dynamic active learning methods should be compared is not the one that uses only randomly selected training pairs, but one that employs the static-active method to overcome the extreme skewness in class distribution that is typical for duplicate detection problems.

4.3 Using weakly-labeled non-duplicates

While the static-active method allows constructing a training set with a significant number of duplicate record pairs, the inverse problem can be encountered in some real-world situations: a "legacy" training set consisting of identified duplicates may be available, while examples of non-duplicates need to be collected. For such situations we consider an unsupervised technique for collecting negative examples. Since duplicate records are rare in a typical database, two randomly selected records are likely to be non-duplicates, and therefore are potentially useful as negative training examples. To help ensure that no duplicate records are included among these pairs, only pairs of records that do *not* share a significant number of common tokens are included as negative examples. Such selection of "weakly-labeled" (and potentially noisy) non-duplicate record pairs is the unsupervised analog of static active selection of duplicates. The process can also be thought of as the opposite of blocking or canopies techniques that use off-the-shelf metrics to avoid comparing "obvious" non-duplicates to speed up the duplicate detection process.

We have conducted experiments where we compared the performance of MARLIN trained using weakly-labeled negatives with experiments where user-labeled negatives were used. Figures 6 and 7 present the results of these experiments on the *Restaurant* and *Cora* datasets. Weakly-labeled negatives were selected randomly from record pairs that shared no more than 20% of tokens to minimize the noise. All experiments used training sets composed of two parts: half the examples were positive examples randomly selected among user-labeled duplicate pairs, and the other half was composed of either weakly-labeled non-duplicate records or randomly selected labeled record pairs. TF-IDF cosine similarity and edit distance with affine gaps were used as the underlying textual similarity metrics for individual fields.

The results again demonstrate that the utility of training-data selection heuristics is dataset-dependent. On *Restaurant*, where duplicate pairs are scarce and randomly selected records are true non-duplicates with very high probability, using a number of "obvious" non-duplicates yields results identical to random selection when a large number of examples is selected, and actually improves slightly over random selection when the training set is small. We conjecture that biasing the SVM with "highly negative" examples when very little training data is available allows learning a better separating hyperplane. On *Cora*, using weakly-labeled negatives leads to slight degradation of system accuracy, which is expected since duplicates are relatively frequent, and noise is likely to be introduced when negative examples are collected in an unsupervised manner. However, the drop in performance is small, and in situations where human labeling of negatives is expensive or infeasible (e.g. due to privacy issues), using weakly-labeled non-duplicates is a viable avenue for unsupervised acquisition of negative examples.

5. FUTURE WORK

It would be interesting to compare static-active sample selection with active learning techniques [16, 18], since the results would help determine how much active learning helps select informative training pairs beyond that of just balancing the extremely uneven class distribution in the training data. The proposed scheme for weakly-labeled negative selection results in training sets that never contain "near-miss" negative examples, since the inverse blocking method guarantees that records selected for negative pairs are dissimilar. While there were no indications that this methodology hurts accuracy compared to random selection of true negative examples, it would be interesting to compare weakly-labeled negative selection to active learning to determine how much accuracy improvement can be obtained from employing near-miss negatives during training. Comparing active learning with a combination of the two techniques that we proposed, static-active duplicate selection and weakly-labeled non-duplicate selection, could also yield interesting experimental results. Finally, exploring ap-

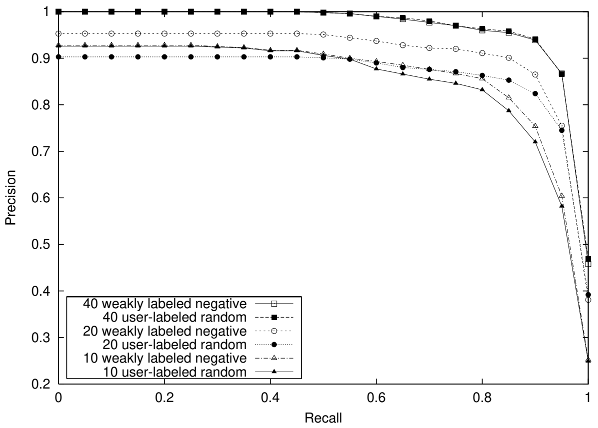

Figure 6: Comparison of using weakly-labeled non-duplicates with using random labeled record pairs on the *Restaurant* dataset.

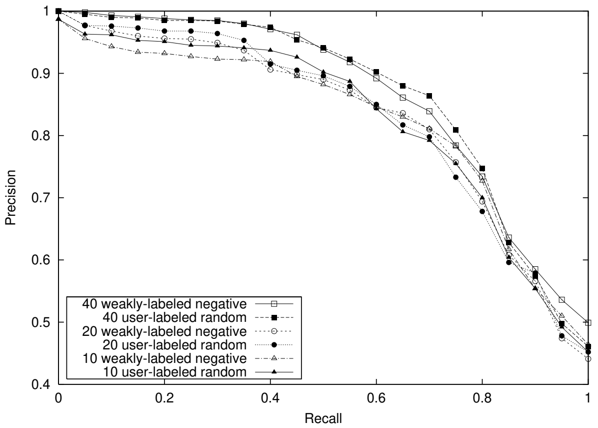

Figure 7: Comparison of using weakly-labeled non-duplicates with using random labeled record pairs on the *Cora* dataset.

proaches to using weakly-labeled record pairs as both duplicate and non-duplicate training examples could potentially lead to new purely unsupervised duplicate detection methods.

6. CONCLUSION

This paper has discussed several important issues in evaluation and training-set construction for duplicate detection. First, research in the area would benefit from a uniform experimental methodology, therefore, we propose precision-recall curves as the most appropriate methodology to adopt. Second, the deduplication accuracy of adaptive systems depends on the similarity between training and test data. We have explored the effect of moving from a "consulting" approach using training data from the same database to a "shrink-wrap" approach using training data from other databases of decreasing similarity. Finally, we have explored the effect of using various approaches to collecting labeled training data, introducing two new approaches: static-active learning and weakly-labeled non-duplicates, and presenting experimental results demonstrating their effectiveness.

7. ACKNOWLEDGMENTS

We would like to thank William Cohen for providing us the *Cora* dataset, Sheila Tejada for the *Restaurant* dataset, and Steve Lawrence for providing the *Citeseer* datasets. This research was supported by the National Science Foundation under grant IIS-0117308 and a Faculty Fellowship from IBM Corporation.

8. REFERENCES

[1] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *Proceedings of VLDB-2002*, Hong Kong, China, 2002.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, New York, 1999.

[3] T. R. Belin and D. B. Rubin. A method for calibrating false-match rates in record linkage. *Journal of the American Statistical Association*, 90(430):694–707, 1995.

[4] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of ACM SIGKDD-2003*, Washington, DC, 2003.

[5] W. W. Cohen and J. Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of ACM SIGKDD-2002*, Edmonton, Alberta, 2002.

[6] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969.

[7] Y. Freund and L. Mason. The alternating decision tree learning algorithm. In *Proceedings of ICML-1999*, Bled, Slovenia, 1999.

[8] D. Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, New York, 1997.

[9] M. A. Hernández and S. J. Stolfo. The merge/purge problem for large databases. In *Proceedings of ACM SIGMOD-95*, pages 127–138, San Jose, CA, May 1995.

[10] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press, 1999.

[11] S. Lawrence, K. Bollacker, and C. L. Giles. Autonomous citation matching. In *Proceedings of the Third International Conference on Autonomous Agents*, New York, NY, May 1999. ACM Press.

[12] A. K. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of ACM SIGKDD-2000*, pages 169–178, Boston, MA, Aug. 2000.

[13] A. E. Monge and C. Elkan. The field matching problem: Algorithms and applications. In *Proceedings of KDD-96*, pages 267–270, Portland, OR, Aug. 1996.

[14] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *Advances in Neural Information Processing Systems 15*. MIT Press. 2003.

[15] F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of ICML-98*, Madison, WI, 1998.

[16] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of ACM SIGKDD-2002*, Edmonton, Alberta, 2002.

[17] S. Tejada, C. A. Knoblock, and S. Minton. Learning object identification rules for information integration. *Information Systems Journal*, 26(8):635–656, 2001.

[18] S. Tejada, C. A. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of ACM SIGKDD-2002*, Edmonton, Alberta, 2002.

[19] W. E. Winkler. Advanced methods for record linkage. Technical report, Statistical Research Division, U.S. Census Bureau, Washington, DC, 1994.

[20] W. E. Winkler. Methods for record linkage and Bayesian networks. Technical report, Statistical Research Division, U.S. Census Bureau, Washington, DC, 2002.