2008-12-11

Learning-based Fusion for Data Deduplication

Sabra Dinerstein
sabra.dinerstein@gmail.com

Parris K. Egbert
egbert@cs.byu.edu

Stephen W. Clyde

Jared Dinerstein

Original Publication Citation

Jared Dinerstein, Sabra Dinerstein, Parris K. Egbert, Stephen W. Clyde. "Learning-based Fusion for Data Deduplication", In Proceedings of The Seventh International Conference on Machine Learning and Applications, pp. 66 - 71, 28. IEEE Computer Society.

2008 Seventh International Conference on Machine Learning and Applications

Learning-based Fusion for Data Deduplication

Jared Dinerstein
Utah State University
Logan, UT 84322
jdinerstein@gmail.com

Sabra Dinerstein
Brigham Young University
Provo, UT 84602
sabra.dinerstein@gmail.com

Parris K. Egbert
Brigham Young University
Provo, UT 84602
egbert@cs.byu.edu

Stephen W. Clyde
Utah State University
Logan, UT 84322
swc@mdsc.com

Abstract

Rule-based deduplication utilizes expert domain knowledge to identify and remove duplicate data records. Achieving high accuracy in a rule-based system requires the creation of rules containing a good combination of discriminatory clues. Unfortunately, accurate rule-based deduplication often requires significant manual tuning of both the rules and the corresponding thresholds. This need for manual tuning reduces the efficacy of rule-based deduplication and its applicability to real-world data sets. No adequate solution exists for this problem.

We propose a novel technique for rule-based deduplication. We apply individual deduplication rules, and combine the resultant match scores via learning-based information fusion. We show empirically that our fused deduplication technique achieves higher average accuracy than traditional rule-based deduplication. Further, our technique alleviates the need for manual tuning of the deduplication rules and corresponding thresholds.

1 Introduction

In today's information technology-driven world, it is imperative that the data found in one's database be both reliable and accurate. Unfortunately, mistakes are common, and these mistakes can reduce the reliability and quality of stored data. Data entry errors, missing integrity constraints, and mismatched data types, are just a few of the possible causes of the loss of data quality [5]. One way to improve the data quality of a database is through *data deduplication*, which is the process of identifying duplicate records (i.e., records that correspond to the same real-world entity).

Rule-based deduplication is a popular technique for identifying duplicate data records [5] [16]. Rule-based systems utilize expert domain knowledge to perform deduplication. Such rule-based systems can achieve high deduplication accuracy. However, achieving high accuracy in a rule-based deduplication system requires not only the creation of relevant rules, but also the tuning of those rules for the current domain and data set [5]. Unfortunately, it can require significant manual effort to adequately tune the rules to achieve acceptable deduplication accuracy. The need for extensive manual tuning directly reduces the applicability of rule-based deduplication to real-world data sets.

We present a novel technique for rule-based deduplication that utilizes match score-level fusion [8] instead of manual rule tuning. We apply our novel two-stage rule-based deduplication technique to biographic text records that comply with the Global Justice XML Data Model (GJXDM) schema [6]. We calculate the individual deduplication rule match scores; these individual rule match scores are then *fused* (i.e., combined) intelligently, using an appropriately trained Support Vector Machine (SVM). We show empirically that our fused deduplication technique achieves higher average accuracy than traditional (non-fused) rule-based deduplication, and demonstrate that our fused deduplication technique alleviates the need for manual rule tuning.

2 Related Work

Data deduplication (i.e., *record linkage* or *record matching*) is not a new problem [5] [7] [11] [17]. Variations in record data can be caused by a number of factors, including: typographical mistakes, intentional misinformation, multiple enrollment, missing or unknown data, and the inclusion of data from disparate sources [4] [5].

Several approaches exist for matching text data records

978-0-7695-3495-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ICMLA.2008.83

[5]. Character-based matching techniques act very locally, on single characters or short character sequences [17]; these techniques employ metrics such as edit distance [9] and affine gap [14]. Q-grams extend character-based similarity metrics to short q-length sequences of characters [15]. Larger-scale similarity information can be measured via token-based techniques, which often utilize *atomic strings* [10] and *tf.idf* calculations [4]. Text-based deduplication is further aided by the use of phonetic similarity measures (such as Soundex [12]), which have proven useful for matching proper names.

Rule-based similarity techniques take advantage of expert domain knowledge [16], and allow the deduplication efforts to be tuned to the current data set. The accuracy of rule-based deduplication is directly dependent on the set of rules that are used, and often requires many iterations of tuning, including the selection of appropriate clue- and rule-level thresholds [5].

Learning-based techniques have recently been applied to text deduplication. Active learning [13], genetic programming [2], and Expectation-Maximization [1] algorithms have all been employed in order to learn specialized distance functions that yield good deduplication results. Additionally, an SVM has been used to combine the individual match results of all text fields contained in a data record [1].

However, learning-based fusion has not yet been applied to rule-based deduplication systems. Particularly, no one has yet applied match score-level fusion [8] to the individual rules that comprise a rule-based deduplication system. The previous work in learning-based deduplication lends credence to our approach, but none of this previous work reduces the burden of manual rule tuning.

Contributions We present a novel technique for rule-based deduplication. We perform match score-level fusion [8] on the output of individual deduplication rules. We show empirically that our fused deduplication technique achieves high average accuracy and removes the need for extensive manual tuning of the deduplication rules and thresholds.

Our fused deduplication technique can be applied to any deduplication problem domain. In this paper, we demonstrate our technique by introducing novel deduplication rules for the GJXDM schema [6]. This schema is a standard for the storage and communication of biographic and biometric data between U.S. government agencies. Government databases (such as state and national census databases) are frequently very large and contain many inexact duplicates [17]. Deduplication of such databases is a significant real-world problem for which there is no adequate solution; thus our contributions include the creation of an accurate deduplication system for the GJXDM schema.

```
<PersonNameType>
    <PersonGivenName> Robert </PersonGivenName>
    <PersonSurName> Jones </PersonSurName>
    <PersonNameSoundexText> R163 J520 </PersonNameSoundexText>
</PersonNameType>
```

Figure 1. *GJXDM XML fragment*. The GJXDM schema is a standard U.S. government schema, containing both biographic and biometric elements. The XML fragment shown here contains data for the surname, the given name, and the Soundex [12] code of both names.

```
TwoStageDeduplication (p) {   //p is a probe record

    primaryCandidateSet = PrimaryRule1(p)

    if (primaryCandidateSet is not empty)
        matches = SecondaryRules(p, primaryCandidateSet)
            if (matches.bestScore > 0)
                Return matches

    primaryCandidateSet = PrimaryRule2(p)

    if (primaryCandidateSet is not empty)
        matches = SecondaryRules(p, primaryCandidateSet)
            if (matches.bestScore > 0)
                Return matches
    ...
}
```

Figure 2. *Two-stage, rule-based deduplication*. The primary rules (applied in decreasing order of importance) create the *primary candidate set*. The secondary rules are only applied to the primary candidate set, rather than to the entire data set.

3 Deduplication Technique

We perform deduplication on records containing GJXDM biographic text data (e.g., given name, surname, date of birth, and affiliations) [6]. (See Figure 1 for an example of the GJXDM schema.) In parallel, we apply each individual deduplication rule to this biographic text data, resulting in a (per rule) set of suspected duplicates and their associated match scores – one set of suspected duplicates per rule. After separately applying all of the deduplication rules, we fuse the resultant rule-level match scores, to produce one overall match decision. This match score-level fusion [8] alleviates the need for manual tuning of the deduplication rules and the rule-level thresholds.

We present our fused deduplication technique in more detail in the following sections.

3.1 Rule-based Deduplication

Our fused deduplication technique can be applied to any rule-based deduplication system, as long as that system outputs a real-valued match score for each rule that is employed. To demonstrate our fused deduplication technique,

```
PrimaryRule1 (p) {   //p is a probe record

    create empty candidateList

    for each g in the gallery {
        if (p.passportID == g.passportID first 3 chars)
            if (p.passportCountry == g.passportCountry first 3 chars)
                if (p.visaID == g.visaID first 3 chars)
                    if (p.visaType == g.visaType first 3 chars)
                        add g to candidateList
    }

    Return candidateList
}
```

Figure 3. *Primary Rule 1*. Primary Rule 1 is the strictest of our four primary rules, and relies on strongly identifying fields such as *passportID*.

```
PrimaryRule2 (p) {   //p is a probe record

    create empty candidateList

    for each g in the gallery {
        if (p.passportID == g.passportID first 3 chars)
            if (p.gender == g.gender)
                if (p.nationality == g.nationality first 3 chars)
                    if (p.dateOfBirth exists)
                        if (g.dateOfBirth exists)
                            add g to candidateList
    }

    Return candidateList
}
```

Figure 4. *Primary Rule 2*. This primary rule is applied *only* if the first primary rule does not return any candidates. More lenient than the first primary rule, Primary Rule 2 does not require a match in as many strongly identifying fields.

```
PrimaryRule3 (p) {   //p is a probe record

    create empty candidateList

    for each g in the gallery {
        if (p.gender == g.gender)
            if (p.nationality == g.nationality first 3 chars)
                if (p.surname == g.surname first char)
                    if (p.givenName == g.givenName first char)
                        if (p.orgName == g.orgName first char)
                            add g to candidateList
    }

    Return candidateList
}
```

Figure 5. *Primary Rule 3*. This rule is even more lenient than the previous primary rules: for most fields, we only require a match in the first character.

```
PrimaryRule4 (p) {   //p is a probe record

    create empty candidateList

    for each g in the gallery {
        if (SoundexDistance(p.surname, g.surname) ≥ 0.75) {
            if (SoundexDistance(p.givenName, g.givenName) ≥ 0.75)
                add g to candidateList
        }
    }

    Return candidateList
}
```

Figure 6. *Primary Rule 4*. Primary Rule 4 is the loosest of our four primary rules. This rule uses Soundex [12] to phonetically compare proper names, which are known to be ambiguous fields. We compare the proper names, in both forward and backward directions, in order to reduce the dependency on the first character of each name.

we have created a two-stage rule-based deduplication system for the GJXDM schema [6]. The two stages of our deduplication system are applied in a waterfall fashion, and are comprised of:

1. a set of computationally inexpensive *primary rules*, and
2. a set of more expensive *secondary rules*.

See Figure 2 for a pseudo-code description of our two-stage, rule-based deduplication system.

3.1.1 Primary Rules

The purpose of our primary rules is to reduce the overall problem size without expending an undue amount of computational effort. As described in Figure 2, the primary rules produce an initial set of probable matches, which we denote the *primary candidate set*. The secondary rules are applied only to this primary candidate set. Thus, the primary rules perform *blocking* (i.e., *search space reduction*) for the secondary rules.

We created four primary rules, summarized in Figures 3 - 6. These primary rules are applied in decreasing order of importance, where Primary Rule 1 is the strictest and most important of our rules. A *strict rule* relies on *strongly identifying* fields, such as passport and visa numbers. A *lenient rule* is comprised of more ambiguous fields, such as surname and given name. Thus, the stricter primary rules tend to produce a smaller primary candidate set than do the lenient rules.

Note that search space reduction can increase the False Reject Rate (FRR) of a system: constraining the search space increases the possibility of skipping an actual match. In order to maintain an FRR $\leq$ 1%, we employed a distinctly lenient rule, Primary Rule 4 (summarized in Figure 6), which utilizes Soundex to phonetically compare proper names [12].

We apply the primary rules in a short-circuited fashion: if a primary rule returns any candidates, the secondary rules

```
SecondaryRule1 (p, c) {   //p is a probe, c is a candidate match
    matchScore = 0

    if (EditDistance (p.passportID, c.passportID) ≤ 1)
        matchScore += 10  //clue weight
    if (EditDistance (p.passportCountry, c.passportCountry) ≤ 1)
        matchScore += 5
    if (EditDistance (p.visaID, c.visaID) ≤ 1)
        matchScore += 8
    if (EditDistance (p.visaType, c.visaType) ≤ 1)
        matchScore += 2

    if (matchScore ≥ 20)  //rule-level threshold
        return (matchScore/25)
    else
        return 0
}
```

Figure 7. *Secondary Rule 1*. Our secondary rules calculate the record match scores. Here, we calculate the Levenshtein edit distance of identifying fields, or *clues*, such as *passportID*. Clues are weighted by their discriminatory power, and only those match scores that meet the rule threshold are returned.

```
SecondaryRule2 (p, c) {   //p is a probe, c is a candidate match
    matchScore = 0

    if (EditDistance (p.passportID, c.passportID) ≤ 1)
        matchScore += 8  //clue weight
    if (p.gender == c.gender)
        matchScore += 3
    if (EditDistance (p.surname, c.surname) ≤ 2)
        matchScore += 6
    if (EditDistance (p.nationality, c.nationality) ≤ 1)
        matchScore += 3
    if (EditDistance (p.dateOfBirth, c.dateOfBirth) ≤ 1)
        matchScore += 3

    if (matchScore ≥ 17)  //rule-level threshold
        return (matchScore/23)
    else
        return 0
}
```

Figure 8. *Secondary Rule 2*. This rule utilizes more clues than does Secondary Rule 1. The selection (and total number) of clues in each rule is based on the discriminatory power of the fields that comprise each clue.

are immediately applied. Subsequent primary rules are only applied if the secondary rules do not return any matches.

Table 1 lists the deduplication accuracy that is achieved by each primary rule. The stated values represent the overall match accuracy that is achieved by applying the specified rule after applying all previous primary rules. The deduplication accuracy values shown in Table 1 could be improved by iterative manual tuning of the rules and their corresponding thresholds [5]. We purposefully did not finely tune the deduplication rules. Instead, we employ learning-based fusion to alleviate the necessity of manual rule tuning.

3.1.2 Secondary Rules

Our secondary rules, shown in Figures 7 - 9, produce the final deduplication results: match scores are calculated, and any duplicate records are flagged. See Table 2 for a summary of the deduplication accuracy that is achieved by each secondary rule (applied in decreasing order of importance).

As described in Figures 7 - 9, our secondary rules utilize identifying fields, or *clues*, such as *passportID*. We weight each clue by the discriminatory power of its underlying field(s). For example, as shown in Figure 7, *passportID* is given a higher weight than is *passportCountry*. This is to be expected: *passportID*, which represents the specific passport number, is a stronger identifier than the passport's issuing country. In other words, while many passports are issued by the same country, no two passports from the same country should share the same passport number.

We measure clue similarity via the Levenshtein edit distance [9]: if the edit distance is within the specified *allowable edit distance* then the weight of that clue is added to the record match score. If the record match score meets the threshold for that rule, the record is flagged as a match. Our rule-level thresholds are based on the discriminatory power of the clues that are included in each rule: if a rule contains ambiguous fields, we allow a larger edit distance and apply a looser overall rule threshold.

As noted previously, rule-based deduplication systems are typically very accurate because they can be tuned to the current domain and data set, but this accuracy comes at the cost of manual tuning [5] [16]. To demonstrate that our fused deduplication technique is not dependent on extensive manual tuning, we purposefully refrained from tuning the clue- and rule-level thresholds.

3.2 Rule Match Score-Level Fusion

Our fused deduplication technique proceeds in much the same manner as the non-fused rule-based deduplication system presented in Section 3.1. However, our fused deduplication technique does not rely on manually tuned rule-level thresholds. Instead, we employ learning-based fusion [8] to determine the appropriate rule-level thresholds, based on both the current data set and the discriminatory power of the individual rules.

1. First, we calculate the *single rule* match scores.

   In parallel, we apply each secondary rule: the current probe record is compared against each primary candidate record, resulting in a set of candidate match scores for each rule. Unlike our traditional (non-fused) rule-based deduplication system (as described in Section 3.1), no rule-level thresholds are employed. Instead, every match score is returned.

```
SecondaryRule3 (p, c) {   //p is a probe, c is a candidate match
    matchScore = 0

    if (p.gender == c.gender)
        matchScore += 3  //clue weight
    if (EditDistance (p.nationality, c.nationality) ≤ 1)
        matchScore += 3
    if (EditDistance (p.surname, c.surname) ≤ 2)
        matchScore += 7
    if (EditDistance (p.givenName, c.givenName) ≤ 2)
        matchScore += 5
    if (EditDistance (p.orgName, c.orgName) ≤ 2)
        matchScore += 8

    if (matchScore ≥ 20)  //rule-level threshold
        return (matchScore/26)
    else
        return 0
}
```

Figure 9. *Secondary Rule 3*. This rule contains ambiguous fields, such as proper names. Thus we apply slightly looser clue thresholds (i.e., larger allowable edit distances) to this rule, as compared to those used in Secondary Rule 1.

2. Next, we perform fused *multi-rule* deduplication, using an appropriately trained SVM.

   We fuse the candidate match scores that are calculated by the individual rules. The fusion SVM takes the single rule match scores as input and outputs one overall classification decision, indicating match/no match.

Our fused deduplication technique does not replace the domain expert: we utilize the original deduplication rules, and thus retain the expert domain knowledge that was used to create those rules. However, our fused deduplication technique reduces the manual tuning of those expert-created rules, using an appropriately trained SVM.

In the next section, we describe the training of our fusion SVM.

3.2.1 Training the Fusion SVM

First, we create *single rule* match score examples, by separately applying each secondary rule. (See Figures 7 - 9 for a pseudo-code description of our secondary rules.) We scale the single rule match scores to be in the range [-1, 1], where a value of 1 represents a perfect match [3].

Next, we concatenate *single rule* match score examples of the same class (i.e., match/no match) to create *multi-rule* examples of the form:

(Classification of the *multi-rule* example,
Scaled *Secondary Rule 1* match score,
Scaled *Secondary Rule 2* match score,
Scaled *Secondary Rule 3* match score).

We then input these multi-rule examples into our fusion SVM. The fusion SVM performs match score-level fusion [8] on the single rule match scores, and outputs one overall classification decision. The fusion SVM learns to classify the multi-rule match score vector as: match/no match.

Our fusion SVM is implemented via LIBSVM [3]. Specifically, the fusion SVM employs a Radial Basis Function (RBF) kernel: $e^{-\gamma(|u-v|^2)}$. We choose the appropriate γ-value and constraints-violation cost, C, at run-time by performing k-fold (stratified) cross-validation on the current set of training examples. The γ- and C-values that produce the best cross-validation accuracy are then used to train the SVM on the entire set of training examples.

Fused deduplication training examples are drawn randomly, with replacement, from the set of multi-rule match score examples. Testing examples are selected in the same manner. We explicitly disallow crossover between the training and testing examples.

	Deduplication Accuracy
Primary Rule 1	16.06%
Primary Rule 2	22.34%
Primary Rule 3	44.18%
Primary Rule 4	72.71%
All Primary Rules	72.71%

Table 1. *Deduplication accuracy of each primary rule.* We apply our primary rules in decreasing order of importance. The values shown here indicate the total accuracy achieved by the specified primary rule, after applying all previous primary rules.

	Deduplication Accuracy
Secondary Rule 1	22.81%
Secondary Rule 2	23.20%
Secondary Rule 3	72.71%
All Secondary Rules	72.71%

Table 2. *Deduplication accuracy of each secondary rule.* The values shown here indicate the total accuracy achieved by the specified secondary rule, after applying all previous secondary rules. Here, the secondary rules are applied to the primary candidate set that is created by all four primary rules, together.

4 Empirical Results

Table 3 summarizes the match accuracy achieved by our deduplication system. As can be seen, our fused deduplication technique yields distinctly higher average accuracy than the comparable traditional (non-fused) rule-based deduplication system: 96.69% versus 72.71%. (The fused deduplication accuracy shown in Table 3 was achieved with

	Match Accuracy
Traditional Rule-based Deduplication	72.71%
Fused Deduplication	96.69%

Table 3. *Fused deduplication accuracy.* Our fused deduplication technique yields distinctly higher average accuracy than our traditional (non-fused) rule-based deduplication. The deduplication accuracy values shown here were achieved without any manual tuning of the clue- and rule-level thresholds. This fused deduplication accuracy reflects the use of 450 training examples and 3000 testing examples, and is averaged over 10 runs, with a standard deviation of 0.548.

450 training examples and 3000 testing examples, and is averaged over 10 runs.)

These deduplication accuracy values are specific to the set of deduplication rules that is employed; a different set of rules will yield different accuracy values, for both fused and non-fused deduplication. However, if the deduplication rules produce consistent match scores, and the fusion SVM is trained appropriately, the use of information fusion should increase the deduplication accuracy.

The traditional rule-based deduplication accuracy shown in Table 3 reflects the fact that we did not finely tune the set of rules that is used. Instead, we demonstrate that our fused deduplication technique produces high average accuracy, even when fusing imperfect individual rules. These results imply that our fused deduplication technique alleviates the need for manual rule tuning. Reducing this manual tuning makes rule-based deduplication plausible for real-world data sets [5].

5 Conclusions

We have presented a novel learning-based fusion technique for rule-based deduplication. Our fused deduplication system utilizes the original deduplication rules, thereby making use of the expert domain knowledge that was used to create those rules. Our fused deduplication technique achieves high average accuracy, without requiring extensive manual tuning of the deduplication rules. We have demonstrated the efficacy of our fused deduplication technique on biographic text records that conform to the significant real-world schema, GJXDM [6].

References

[1] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48, New York, NY, USA, 2003. ACM.

[2] M. G. Carvalho, A. H. F. Laender, M. A. Gonçalves, and A. S. da Silva. Replica identification using genetic programming. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 1801–1806, New York, NY, USA, 2008. ACM.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] W. Cohen. Integration of heterogeneous databases without common domains using queries based on textual similarity. *SIGMOD Rec.*, 27(2):201–212, 1998.

[5] A. Elmagarmid, P. Ipeirotis, and V. Verykios. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19:1–16, 2007.

[6] GJXDM. Global Justice XML Data Model. `http://www.it.ojp.gov/jxdm/`.

[7] K. Goiser and P. Christen. Towards Automated Record Linkage. In *Fifth Australasian Data Mining Conference (AusDM2006)*, volume 61 of *CRPIT*, pages 23–31, Sydney, Australia, 2006. ACS.

[8] A. Jain and A. Ross. Multibiometric systems. *Commun. ACM*, 47:34–40, 2004.

[9] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[10] A. Monge and C. Elkan. The Field Matching Problem: Algorithms and Applications. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 267–270, 1996.

[11] H. Newcombe and J. Kennedy. Record linkage: making maximum use of the discriminating power of identifying information. *Commun. ACM*, 5(11):563–566, 1962.

[12] R. Russell. United states patent: 1261167, Apr. 1918.

[13] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–278, New York, NY, USA, 2002. ACM.

[14] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, March 1981.

[15] E. Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theor. Comput. Sci.*, 92:191–211, 1992.

[16] Y. Wang and S. Madnick. The Inter-Database Instance Identification Problem in Integrating Autonomous Systems. In *Proceedings of the Fifth International Conference on Data Engineering*, pages 46–55, Washington, DC, USA, 1989. IEEE Computer Society.

[17] W. Winkler. The state of record linkage and current research problems. Technical Report RR/1999/04, Statistics Research Division, U.S. Bureau of the Census, 1999.