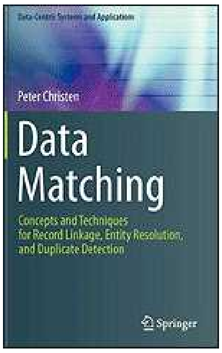

Data-Centric Systems and Applications
Peter Christen
Data Matching
Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection
Springer

Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection

by Peter Christen

Chapter 7: Evaluation of Matching Quality and Complexity

7.1 Overview

Over the past decades, as the previous chapter has shown, various classification techniques for data matching have been developed. The main objective of these techniques is to achieve high matching quality. Similar to other classification problems, in order to be able to assess the quality of the matched data for a certain data matching project, ground-truth data, also known as 'gold standard' data, are required. The characteristics of such ground-truth data must be as close as possible to the characteristics of the data that are to be matched.

To summarise, if a record pair has been classified as a match, then the assumption is that both records in the pair refer to the same real-world entity. For a record pair classified as a non-match, on the other hand, the two records in the pair are assumed to refer to two different real-world entities. Thus if a ground-truth data set with known true matching and non-matching record pairs is available, then similar to other classification problems in machine learning and data mining [135], a variety of measures can be calculated on the outcomes of the classification process. Several such measures are discussed in this chapter.

The question now arises: how to acquire such ground-truth data for a certain data matching exercise. In many if not most data matching situations no ground-truth data are readily available. There are several approaches of how ground-truth data can be generated.

One possibility is that the results from a previous data matching project in the same domain (ideally an earlier version of the same databases) are available, and that these databases have been manually evaluated with regard to the quality of the previous matching outcomes. For example, domain experts might have detected wrongly matched as well as missed true matching pairs of records as they have worked with the matched databases. The quality of previously matched data might however not be good enough to be used as training data, especially if a more simpler matching approach was previously employed. Additionally, the manual inspection and possible correction of matches are often not 100 % correct, and it is therefore likely that the databases used as ground-truth data contain mistakes with regard to the match status of certain record pairs.

Another approach to obtain ground-truth data is to manually generate such data by sampling pairs of records from the two databases that are to be matched (or pairs from the single database that is to be deduplicated), and to manually classify these pairs as being either a match or a non-match. This approach has two difficulties.

The first is similar to the drawbacks described above when a previously matched database is used as ground-truth data, in that the manual classification of record pairs is unlikely to always be correct. Some mistakes will potentially be introduced by a human classification. These mistakes will not be in the record pairs that are easy to classify. Two records that differ in all attribute values are very obviously not a match. Similarly, two records where all attribute values are the same or only contain minor differences can be manually classified as a match with high confidence. These two types of record pairs can also be easily classified automatically, as was discussed in the previous chapter.

However, the record pairs that contain variations or differences in several of their attribute values are hard to classify. These variations include ambiguous names, or changed name or address values that are due to a person having married or moved to a new address. Often, additional information is required so an accurate manual classification can be performed. Section 7.4 will further cover this topic in the context of manual clerical review of potential matches.

The second drawback when manually generating ground-truth data based on sampling record pairs from the databases to be matched is the overall distribution of matches and non-matches in the classified record pairs. Assuming no indexing or blocking (as discussed in Chap. 4) has been applied, the matching of two databases that contain m and n records, respectively, will generate $m \times n$ record pairs that need to be classified. If it is assumed that both databases have been deduplicated prior to the matching, then a maximum of $\min(m, n)$ true matching record pairs are contained in the $m \times n$ record pairs. The number of true matches is therefore much smaller than the number of non-matches, especially as the size of the databases increases. The same holds for the deduplication of a single database that contains m records, where (without indexing) $m(m - 1)/2$ record pairs need to be compared, but where the maximum number of true matches will be $m - 1$ (in the unlikely case where $m - 1$ records are duplicates of one single record). The sizes of the match and non-match classes are therefore often very imbalanced in data matching. Even if some form of indexing has been applied, the number of candidate record pairs generated is very likely to be much larger than the number of true matches contained in them.

Using simple random sampling of candidate record pairs and manually classifying the sampled pairs to generate a ground-truth data set will therefore result in a sampled set that mostly contains non-matching record pairs. A stratified sampling approach can be employed, such that a balanced number of true matches and non-matches is sampled from all record pairs. This can for example be achieved by binning the comparison vectors of record pairs according to their summed similarities, and then sample the same number of comparison vectors from each bin.

A third approach to obtain data that contain the true match status of record pairs is to use one of the small number of publicly available test data sets that have been generated by researchers to test their algorithms. An overview of such data sets is given later in this chapter in Sect. 7.5. Finally, a fourth approach is to use synthetically generated data that have similar characteristics as the real databases that are to be matched. This approach will be discussed further in Sect. 7.6. For these last two approaches, the match status of record pairs is generally known. If it makes sense to use public or synthetic data sets to evaluate a certain data matching system in practice depends upon the actual situation in which a data matching system will be employed.

7.2 Measuring Matching Quality

Assuming some ground-truth data sets with the true match status of all its possible record pairs is available and a matching has been conducted on these data sets, each compared and classified record pair is assigned into one of the following four categories [71]:

- *True positives*. These are the record pairs that have been classified as matches and that are true matches. These are the pairs where both records refer to the same entity.
- *False positives*. These are the record pairs that have been classified as matches, but they are not true matches. The two records in these pairs refer to two different entities. The classifier has made a wrong decision with these record pairs. These pairs are also known as false matches.
- *True negatives*. These are the record pairs that have been classified as non-matches, and they are true non-matches. The two records in pairs in this category do refer to two different real-world entities.
- *False negatives*. These are the record pairs that have been classified as non-matches, but they are actually true matches. The two records in these pairs refer to the same entity. The classifier has made a wrong decision with these record pairs. These pairs are also known as false non-matches.

Figure 7.1 illustrates these four outcomes. The true positives are the intersection of the true matches and classified matches. It is common to illustrate these four possible outcomes of a classification in a confusion or error matrix [135], as shown in Fig. 7.2.

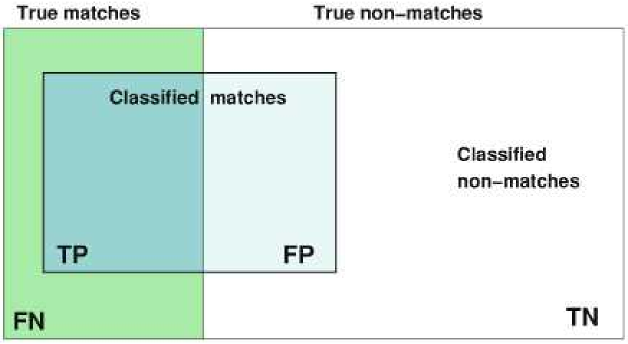


Figure 7.1: Example illustration of the classification outcomes for data matching. *TP* refers to true positives, *FP* to false positives, *TN* to true negatives and *FN* to false negatives, as discussed in Sect. 7.2

		Predicted classes	
		Matches	**Non-matches**
Actual classes	Matches	True Positives (true matches)	False Negatives (false non-matches)
	Non-Matches	False Positives (false matches)	True Negatives (true non-matches)

Figure 7.2: Error or confusion matrix illustrating the outcomes of a data matching classification. As discussed in Sect. 7.2, the aim of the classification is to identify and correctly classify as many true matches as possible while keeping the number of false positives and false negatives as small as possible

As was discussed previously, the number of true negatives in data matching situations will often be much larger than the sum of the number of true positives, false negatives and false positives. The reason for this is the nature of the comparison process, because there are many more pairs where the two records refer to two different entities than there are pairs where both records refer to the same entity [71].

An ideal outcome of a data matching project is to correctly classify as many of the true matches as true positives, while keeping both the number of false positives and false negatives small.

Based on the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), different quality measures can be calculated [71]. The following list presents the measures most commonly used with data matching, and discusses their characteristics and their suitability for assessing the quality of data matching and deduplication.

- *Accuracy*. This quality measure is calculated as

  (7.1)
  $$\mathrm{acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}.$$

  This measure is most widely used for binary as well as multi-class problems in the fields of machine learning and data mining [135]. Accuracy is mainly useful for situations where the classes are balanced, i.e. where the number of instances (record pairs in the case of data matching) are more or less the same for both classes (matches and non-matches).

  As was previously discussed, balanced classes are rare in data matching and deduplication classification, in that the majority of record pairs corresponds to true non-matches (true negatives). As a result, the accuracy measure is not suitable to properly assess matching quality. The value of TN in Eq. 7.1 dominates the calculation of accuracy.

  For example, assume two databases with 1,000,000 records each are matched, and the indexing step resulted in 50,000,000 candidate record pairs that were generated. Now assume that there are 500,000 true matches between these two databases. Also assume a classifier has classified 600,000 record pairs as matches, and of these 400,000 correspond to true matches. As a result, there will be TP = 400,000, FN = 100,000, FP = 200,000 and TN = 49,300,000. The accuracy calculated on these values then is: $\mathrm{acc} = \frac{400{,}000+49{,}300{,}000}{50{,}000{,}000} = 0.994$, which corresponds to 50,000,000 99.4 % accuracy. Clearly, this is not a meaningful measure, because only 400,000 of the 500,000 true matches were classified correctly. Even a simple classification of all candidate record pairs as non-matches (TP = 0, FN = 500,000, and FP = 0) will still result in a very high accuracy value.

  As a result, because of the imbalanced classification problem that data matching and duplication commonly pose, accuracy is not a suitable quality measure and should not be used. The following measures are more appropriate alternatives.

- *Precision*. This is a measure commonly used in information retrieval to assess the quality of search results [288]. It is calculated as

  (7.2)
  $$\mathrm{prec} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}.$$

  Because precision does not include the number of true negatives, it does not suffer from the class imbalance problem in the way accuracy does. Precision calculates the proportion of how many of the classified matches (TP + FP) have been correctly classified as true matches (TP). It thus measures how precise a classifier is in classifying true matches. Precision is also known as the positive predictive value (PPV) in the medical literature [37].

  Using the same numerical values as in the example above, precision can be calculated as: $\mathrm{prec} = \frac{400{,}000}{400{,}000+200{,}000} = 0.667$, which corresponds to a precision of 66.7 %. This means that two-thirds of the record pairs this classifier has classified as matches correspond to true matches, while a third corresponds to false matches.

- *Recall*. This is a second measure that is commonly used in information retrieval It is calculated as

  (7.3)
  $$\mathrm{rec} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

  Similar to precision, because recall does not include the number of true negatives, this measure does not suffer from the class imbalance problem. It measures the proportion of true matches (TP + FN) that have been classified correctly (TP). It thus measures how many of the actual true matching record pairs have been correctly classified as matches. Recall is also known as the *true positive rate* or the *hit rate*, while in the medical literature it is known as *sensitivity* [301] and commonly used to assess the results of epidemiological studies.

  Continuing the numerical example, the recall of this classification outcome can be calculated as:

$\mathrm{rec} = \frac{400{,}000}{400{,}000+100\ 000} = 0.8$, which corresponds to a recall of 400,000+100,000 80.0 %. This means that this classifier has correctly classified four out of every five true matching record pairs.

It should be noted that there is a trade-off between precision and recall. Depending upon the data matching or deduplication situation, it might be more important to achieve matching results with high precision but accept a lower recall, while in other data matching situations having a low precision is acceptable but a high recall is required.

For example, for a crime investigation where certain suspect individuals need to be matched with a large database of people, a high recall is desired to make sure that it is likely that the individuals one is looking for are included in the matched record pairs, even if there is a larger number of matches that need to be investigated. On the other hand, high precision is required in many public health studies where each match would correspond, for example, to a patient with certain medical characteristics who needs to be included into a cohort study. In this situation one wants to be sure to only include patients into the set of matched record pairs who do have the medical condition one is interested in.

- *F-measure*. This measure, also known as *f-score* or *f_1-score*, calculates the harmonic mean between precision and recall [195]:

  (7.4)
  $$\mathrm{fmeas} = 2 \times \left(\frac{\mathrm{prec} \times \mathrm{rec}}{\mathrm{prec} + \mathrm{rec}}\right).$$

  The *f-measure* combines precision and recall and only has a high value if both precision and recall are high. Aiming to achieve a high *f-measure* requires to find the best compromise between precision and recall [15].

  With the continuing numerical example, with prec = 0.667 and rec = 0.8, then $\mathrm{fmeas} = 2 \times (\frac{0.667 \times 0.8}{0.667+0.8}) = 0.727$.

- *Specificity*. This measure is also known as the *true negative rate* and it is commonly used in the medical literature [301]. It is calculated as

  (7.5)
  $$\mathrm{spec} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}.$$

  Because this measure includes the number of true negatives (TN), it suffers from the same problem as accuracy, and should not be used for data matching or deduplication. If the number of false positives (FP) is small compared to the number of TN (which it likely is because of the class imbalance in data matching), then the calculated specificity will be dominated by the number of TN.

  For the numerical example, specificity is calculated as $\mathrm{spec} = \frac{49{,}300{,}000}{49{,}300{,}000+200{,}000} = 0.996$, which corresponds to 99.6 %.

- *False positive rate*. This measure is also known as *fall-out* in information retrieval [288]. It is measured as

  (7.6)
  $$\mathrm{fpr} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}.$$

  Note that fpr = (1 - spec). Because this measure includes the number of true negatives (TN), it suffers from the same problem as accuracy and specificity, and should not be used for data matching.

  Continuing the numerical example, the false positive rate is calculated as $\mathrm{fpr} = \frac{200{,}000}{49{,}300{,}000+200{,}000} = 0.004$, which is a very low 0.4 %. This very low value does not reflect that this classifier classified 80 % of the true matches correctly.

The measure most commonly used in the computer science literature for assessing the quality of data matching has been accuracy [102, 129, 155, 231, 252, 301]. However, precision [29, 83, 185], recall [129, 185, 301] and the f-measure [31, 85, 185] have also been used, and they have gained popularity in recent years as researchers have become more aware of the pitfalls of using the accuracy measure [71].

While the above measures provide a single number of the matching quality achieved by a single classifier, most classification techniques described in Chap. 6 have one or several parameters that can be modified and tuned. Depending upon the value(s) of such parameter(s), a classifier will have a different performance, leading to different numbers of true and false positives and negatives. Rather than a single value for a quality measure using a certain parameter setting, a series of values can be generated for a certain measure using different parameter settings. The resulting values can then be visualised in various ways to illustrate the performance of a certain classifier over a range of parameter settings. Such visualisations also allow more detailed comparisons of the performance of several classification techniques. The following three visualisations (also shown in the example in Fig. 7.3) are commonly used to illustrate the outcomes of the classification of candidate record pairs.

- *Precision–recall graph*. In this visualisation the values of precision and recall are plotted against each other as generated by a classifier with different parameter settings. This type of graph is commonly used in the field of information retrieval to visualise, for example, the quality of results returned by a Web search engine [288]. Figure 7.3b shows an example of a precision–recall graph.

  For each selected parameter setting of a classification model, the precision and recall values are calculated resulting in a single point in the precision–recall graph. Recall is plotted along the horizontal axis (or *x*-axis) of the graph, while precision is plotted against the vertical axis (or *y*-axis). As parameter values are changed, the resulting precision and recall values generally change as well.

  Commonly, a high precision of a classifier will result in a low recall value and vice versa. Therefore, in precision–recall graphs there is often a curve starting in the upper left corner moving down to the lower right corner. Ideally, a classifier should achieve both high recall and high precision and therefore the curve should be as high up in the upper right corner as possible.

- *F-measure graph*. An alternative to plotting two quality measures (such as precision and recall) against each other is to plot the values of one or several measures with regard to the setting of a certain parameter, for example a single threshold used to classify candidate record pairs according to their summed comparison vectors, as was discussed in Sect. 6.2. This is shown in Fig. 7.3c. In this graph, the horizontal axis shows the summed similarity score (SimSum) that is used as classification threshold. For each threshold value, all record pairs with a summed similarity below that threshold will be classified as non-matches and all other pairs as matches. As the threshold is increased from 0 to 3, for this example, the number of record pairs classified as non-matches increases (and thus the number of TN and FN increases), while the number of TP and FP decreases.

  Any of the above discussed quality measures can be plotted in such a graph. An often used combination is to show precision, recall and the f-measure in the same graph, as illustrated in Fig. 7.3c. As the classification threshold is increased, the value of recall gets lower (because less of the true matches, those with a lower overall similarity, are classified as matches), while precision gets higher (because less true non-matches are classified as matches with higher similarity threshold).

- *ROC curve*. Similar to the precision–recall graph, the receiver operating characteristic (ROC) curve is plotted as the values of two quality measures against each other [106]. The horizontal axis is the false positive rate while the vertical axis is the true positive rate (which is the recall). The closer an ROC curve is to the top left corner the better a classifier is, because this means it can achieve a high recall with a small number of false positives.

  While the use of ROC curves is being promoted to be robust against imbalanced classes (as is common in data matching and deduplication) [106], the problem when applying them in data matching is that the number of true negatives, which is a factor only when the false positive rate is calculated, will lead to too optimistic ROC curves because the false positive rate will be calculated to be very low. The use of ROC curves for data matching should therefore be carefully assessed. Plotting several ROC curves generated by different classifiers can certainly help to compare their performance over a range of parameter settings.

  Based on an ROC curve, a numerical measure called the area under the curve (AUC) can be calculated. This is basically the area of an ROC graph that lies in the lower right area of the graph below the curve. The closer an ROC curve is to the upper left corner the larger its AUC value becomes, and therefore the better a classifier performs over a range of parameter values. Note that the value of AUC is always between $0.5 \leq auc \leq 1.0$, because even a random classifier (which would have an ROC curve that is the diagonal in the ROC graph) has an AUC value of $auc = 0.5$, while a perfect classifier will have an AUC of $auc = 1.0$.

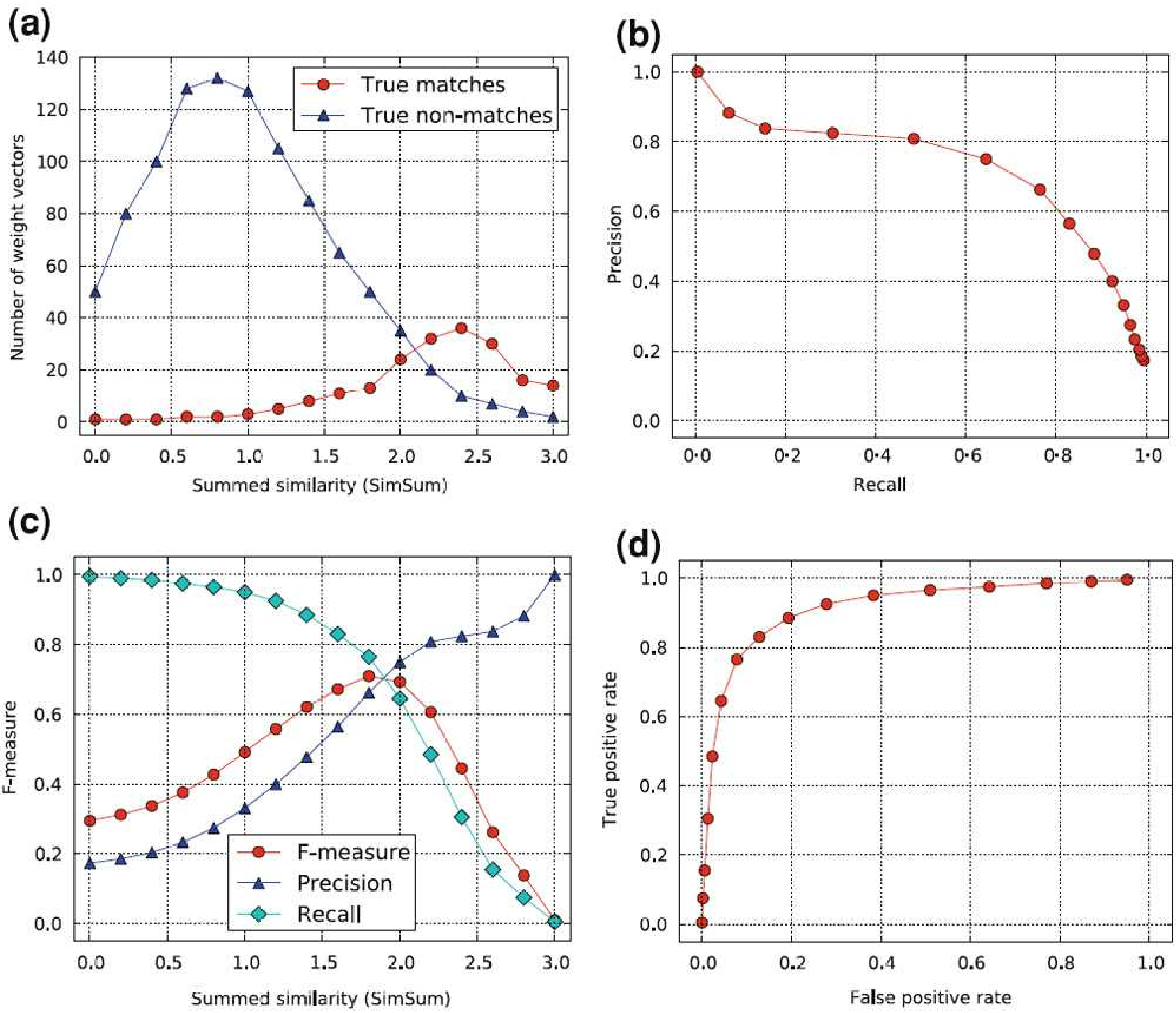


Figure 7.3: An example of simulated classification results assuming 200 true matching and 1000 true non-matching record pairs, and three possible visualisations of the quality results of such a classification. Plot **a** shows the distribution of the comparison vectors summed into similarity values SimSum (as was previously illustrated in Fig. 2.6 on page 31). As discussed in this chapter, the majority of the compared record pairs refer to true non-matches. Plots **b** to **d** show the different quality graphs that can be generated from the summed similarities in Plot **a**. **a** Summed Similarities **b**. Precision-Recall **c**. F-measure **d**. ROC curve

The issue of how to evaluate the merging of matched records into new compound records has recently been investigated by Menestrina et al. [187]. In their work, the authors compare different measures that have been used by researchers in data matching. They show that assessing the outcomes of a data matching project using different measures can lead to different rankings of the matched record pairs, and thus to different matching outcomes. The merging of records into entities is viewed as a clustering process. Each clustering result is compared to the ground-truth data (gold standard). A generalised merge distance (GMD), defined by the authors, is used to assess how close a certain clustering is to the known true clustering of records into entities (where each cluster refers to one entity). The GMD is related to the edit distance, as discussed in Sect. 5.3, in that it assigns costs to merging a cluster of records or splitting a cluster. The smaller the number of merges or splits is from a given set of clusters (the results of a data matching classification) to the true clustering result, the smaller the GMD is. The authors also show that precision, recall and thus the f-measure (Eqs. 7.2–7.4), can all be calculated easily from the GMD.

7.3 Measuring Matching Complexity

Besides the quality of the record pairs classified as matches and non-matches within a data matching or deduplication project, a second major aspect is the efficiency and effectiveness of data matching techniques or systems. One obvious approach would be to simply measure the run-time on different data sets to compare which technique or system is faster. The results of such an assessment would be very specific to the computing hardware used, such as the speed of its processors, and its memory and I/O bandwidths. A platform-independent way to compare systems or techniques would be of advantage because results would be more generalisable. One possibility is to count the number of candidate record pairs generated by an indexing technique, and use this number to calculate a measure of how complex a data matching exercise is. Three such approaches to measuring the efficiency and complexity of data matching and deduplication have been proposed [71].

Following the notation given in previous publications [20, 64, 71, 102, 128], the total number of matched and non-matched record pairs are denoted with n_M and n_N, respectively, with $n_M + n_N = m \times n$ for the matching of two databases that contain m and n records, respectively, and $n_M + n_N = m(m - 1)/2$ for the deduplication of one database that contains m records. Note that these numbers correspond to the full comparison space of all possible record pairs, i.e. when no indexing has been applied. The number of true matched and true non-matched candidate record pairs generated by an indexing technique is denoted with s_M and s_N, respectively, with $s_M + s_N \leq n_M + n_N$. Three measures can now be defined.

- *Reduction ratio*. This measure provides information about how many candidate record pairs were generated by an indexing technique compared to all possible record pairs, without assessing the quality of these candidate record pairs. Reduction ratio is calculated as

  (7.7) $$\mathrm{rr} = 1 - \left(\frac{s_M + s_N}{n_M + n_N}\right).$$

  The reduction ratio therefore measures the relative reduction of the comparison space of a data matching or deduplication exercise. A high reduction ratio means an indexing technique has removed many record pairs from the full comparison space, while a low reduction ratio means that a larger number of candidate record pairs have been generated.

- *Pairs completeness*. This measure takes the true match status of candidate record pairs into account. It is calculated as

  (7.8) $$pc = \frac{s_M}{n_M}.$$

  Pairs completeness corresponds to recall (Eq. 7.3) discussed previously. It is the number of true matching record pairs that have been generated by an indexing technique divided by the total number of true matching pairs in the full comparison space. The lower the pairs completeness value is the more true matches have been removed by an indexing technique. This leads to lower matching quality, because the record pairs removed in the indexing step are implicitly classified as non-matches without being compared in detail.

  There is a trade-off between reduction ratio and pairs completeness [20], i.e. between the number of record pairs that are removed in the indexing step and the number of missed true matches. No indexing technique is perfect in only removing record pairs that correspond to non-matches. Some record pairs that correspond to true matches are likely removed as well in the indexing step. Using an indexing technique that has a lower reduction ratio will mean that a smaller number of candidate pairs is removed in the indexing step. This can often lead to an increased pairs completeness value.

- *Pairs quality*. This third measure, which also takes the quality of candidate record pairs into account, is calculated as

  (7.9) $$\mathrm{pq} = \frac{s_M}{s_M + s_N}.$$

  It is the number of candidate record pairs that correspond to true matches that were generated by an indexing technique, divided by the total number of candidate record pairs that were generated. It corresponds to the measure precision (Eq. 7.2) presented previously. A high pairs quality value means that an indexing technique is successful in generating candidate record pairs that mostly correspond to true matches, while keeping the number of candidate pairs that correspond to non-matches low. Similar to the trade-off between precision and recall, there is normally a trade-off between pairs completeness and pairs quality. Aiming for an increase in one of these two measures generally results in a decrease in the value of the other measure.

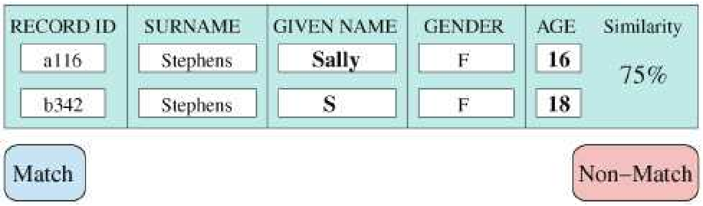
None of these three measures is taking any computational resources into account, such as processing time or main memory usage. These are dependent upon the actual implementation of a data matching system and the computing platform used.

Similar to all quality measures discussed in Sect. 7.2 above, being able to calculate both the pairs completeness and pairs quality measures does require knowledge about the true match status of record pairs. If this information is not available for a given data matching or deduplication exercise, then only the reduction ratio, as well as run-time and memory usage, can be reported. This prohibits a proper assessment of a data matching technique or system.

7.4 Clerical Review

The traditional classification model of probabilistic record linkage (discussed in Sect. 6.3) that has been used in data matching for several decades (and that is implemented in various data matching systems) classifies the compared candidate record pairs into matches, non-matches, as well as potential matches, as Fig. 2.1 on page 24 shows. The class of potential matches consists of record pairs where a decision model was not able to make a clear decision on if they correspond to matches (where both records refer to the same entity) or non-matches (where the two records refer to two different entities) [129].

A manual classification is required for candidate record pairs that have been inserted into the class of potential matches. This manual classification requires a clerical review where each record pair is assessed visually and a match decision is made manually.Figure 7.4 provides an example that illustrates how a pair of records might be presented to a person who conducts such a manual review. Smith and Newcombe in an early study using health records showed that a computer-based probabilistic data matching system can result in more reliable, consistent and more cost effective matching results compared to a fully manual approach [241, 242].



RECORD ID	SURNAME	GIVEN NAME	GENDER	AGE	Similarity
a116	Stephens	**Sally**	F	**16**	75%
b342	Stephens	**S**	F	**18**	

Figure 7.4: Example for a clerical review of a record pair classified as a potential match with an overall similarity calculated to be 75 %. The fields (or attributes) where values are different are shown with a larger font to grab the attention of a reviewer. The 'Match' and 'Non-Match' buttons allow a reviewer to make their manual classification decision

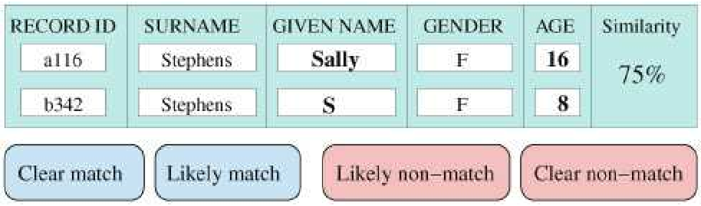
The manual clerical review of potential matches can be a tedious, time-consuming and labour intensive process, especially in cases where the matching of two databases has resulted in a large number of record pairs that were classified as potential matches. This can either be because the databases were large, the data were difficult to classify, or the classification model was not able to accurately discriminate matches from non-matches.

Several aspects make manual clerical review a difficult process. First of all, looking at Fig. 7.4, one can see that even for an experienced domain and data matching expert it can be difficult to make an accurate manual classification when assessing a single record pair in isolation. Other records might be similar and have characteristics that also make them potential match candidates even though they have a lower overall similarity. For the given example, there might be another record with surname 'Stevens', given name 'Sal' and age '17' but with a missing gender value, and an overall similarity of 72 %. This could well be a better matching record. Ideally, therefore, a system that facilitates manual clerical review should visualise not just a pair of records but a whole group of similar records, for example in the same way as a Web search engine presents a list of query results ranked according to relevance. Alternatively, having access to external data that can help validate if a pair of records corresponds to a match or non-match can be highly beneficial in the manual decision-making process. Such external data can for example be a database that contains the known previous addresses or telephone numbers of individuals, or their known nicknames and previous surnames.

A second issue that makes clerical review a difficult process is that the manual match or non-match decision made can differ not only from reviewer to reviewer, but even the same reviewer might make different decisions depending upon their mood, time of day and concentration level. The same reviewer in the morning might classify a pair as a match, but if they would see the same pair late in a day's work might decide it is a non-match based on their mood and desire to finish a day's work. It is therefore of advantage to have more than one reviewer assessing the same set of record pairs, so that in case of a conflicting classification of a certain pair an additional review can be asked for. This of course prolongs the review process and also makes it more costly. As with many aspects of data matching and deduplication, the application of where the matched records will be used dictates how accurate the matched data need to be and how costly a false match or a false non-match will be.

One advantage of the clerical review process is that it can help generate training data of record pairs that are difficult to classify. Such manually generated training examples can flow back into the classification step as illustrated in Fig. 2.1 on page 24. It is however important to be aware of the above discussed issues, and that the class of potential matches does contain the most difficult to classify record pairs. The confidence one can have on the manually classified record pairs depends upon the thoroughness of the manual review process, the system used to present potentially matched pairs to the reviewer, the domain expertise of the reviewers, and if they had access to any external data that supported their decisions. Using manually classified pairs as training data for a record pair classifier therefore needs to be considered carefully.

One way to reduce the possibly large number of potential matches that need to be classified manually is to employ an active learning approach as was discussed in Sect. 6.7. With active learning, only a small number of hard to classify record pairs are manually classified in each iteration, and a new classifier is trained using training data that include these manually classified pairs. The same careful consideration as mentioned before about the confidence one has into the manually classified pairs needs to be considered when active learning techniques are employed. Ideally, an active learning classifier should be able to take a confidence level of the manually classified record pairs into account, as illustrated in Fig. 7.5.



RECORD ID	SURNAME	GIVEN NAME	GENDER	AGE	Similarity
a116	Stephens	**Sally**	F	**16**	75%
b342	Stephens	**S**	F	**8**	

Figure 7.5: A variation of Fig. 7.4 that allows a reviewer to provide feedback about the confidence of their manual classification decision

7.5 Public Test Data

As discussed in Sect. 7.2, knowing the true match status of record pairs is a requirement for being able to measure the matching quality of a data matching or deduplication system for a certain data set. For many real-world applications, however, it is very difficult to obtain or create such ground-truth data. Even if significant manual resources are put into a manual training process (which is similar to the manual clerical review process described in the previous section), then for many of the record pairs added to the training set the match status might not be known with high confidence.

For researchers, who might be working in an academic environment without close collaboration with an organisation that has real data and that is prepared to provide these data for research, it is generally very difficult to get access to any real-world data sets.

As an alternative, researchers have investigated public sources of data that can be used to test and evaluate data matching and deduplication techniques. Because of the private nature of personal information (such as names, addresses, date of birth and so on), data that contain information other than personal details are commonly used by researchers. The issues involved with privacy in data matching are covered further in Chap. 8.

Over the years, a collection of test data sets have been used by various researchers in the field of data matching. The most common of these data sets are described in the following list:

- *Cora*. This data set contains 1,295 bibliographic records of machine learning articles that correspond to 189 actual real publications. A total of 17,184 out of 1,295 × 1,294/2 pairs of records correspond to true matches (assuming no indexing is applied). Each record in this data set contains the publication name, the publication year, one or more author names (sometimes only surnames and initials) and the conference or journal name (or their abbreviation only) where an article was published.

- *Restaurant*. This small data set contains 864 records of restaurant names, addresses, telephone numbers and food style (French, Italian, Japanese, etc.) taken from the Fodor and Zagat restaurant guides. In total, 752 different restaurants are included in this data set, and there are 112 restaurants that appear twice.

- *Census*. This is a pair of small data sets that contain synthetic census records generated by the US Census Bureau. The first data set contains 449 records while the second data set contains 392 records. The number of matching records is 327. Each record consists of the attributes first and last name, middle initials, a street number and a street name.

- *UCD people*. This is a data set which contains the names of people working at the University College in Dublin. Each person is represented by a single string that contains the person's given name and surname, as well as optional titles and a role or position description such as 'Head of Department' or 'Newman Scholar'. Each individual is assigned a unique identifier.

- *CDDB*. This data set consists of 9,763 records with details of compact disc albums (CDs), such as their artist, title, genre and the year when the CD was published. These records were randomly selected from the FreeCD database [195]. A time-consuming manual process lead to the detection of 298 true duplicates in this data set (with a total of 607 true matching record pairs, assuming no indexing is applied). Each unique CD was given a unique identifier.

- *DBLP*. This online database containing computer science journals and conference and workshop proceedings with over a million articles has been used by several data matching research groups for their experimental studies. Each record consists of an article's name, details of the publication venue, its year of publication and the names of the author(s) of that publication. An XML version of this database[1] can be freely downloaded. A drawback of this database is however that the true match status is not known, and therefore it is difficult to use this database to evaluate the accuracy of data matching techniques without some initial processing and generation of some form of ground-truth data.

- *IMDB*. The Internet movie database[2] is another popular source of data used to evaluate data matching and deduplication techniques. The database contains details about different types of entities, such as people (actors, producers, directors, etc.), movies, companies, as well as movie ratings and plot descriptions. Similar to DBLP, no ground-truth data are available. However, compared to DBLP, where it is likely that duplicate records exist for the same article, in the IMDB database it is more likely that several records with the same name refer to different individuals (there are, for example, more than 20 people with the name 'Bill Murray' listed in IMDB), or that several movies have the same name. Researchers commonly corrupt the data they download from IMDB to generate duplicate records. This approach to artificially generating data is discussed further in the next section.

The Cora, UCD people, Restaurant and Census data sets are available in several repositories and open source data matching systems, including the RIDDLE repository[3] the SecondString toolkit,[4] and the FEBRL system.[5] Note that there are various versions of the Cora data set in different repositories. Some have been pre-processed and cleaned further than the original version. The CDDB and Cora data sets (as well as some other useful data sets) are available from the repository maintained by Naumann[6] [195]. Koepcke et al. [169] recently conducted an evaluation of several data matching systems using a set of four real data sets which the authors have made publicly available as part of their framework for evaluation data matching system, FEVER.[7]

It is important to note that these data sets provide only limited amount of information about the performance (with regard to matching quality and complexity) of data matching and deduplication systems or techniques. First, the data sets where the match status is known are all fairly small, therefore limiting the evaluation of scalability of a techniques or system. Second, each of these data sets contains a very specific type of data. Any results achieved on them should not be generalised to other types of data, even to data from the same domain but with different characteristics (such as different error characteristics or different attributes).

A comparison of different indexing techniques on three of the above listed data sets (Cora, Restaurant, and Census) is presented in Sect. 4.10, based on experiments recently presented by Christen [64]. As this evaluation illustrates, different indexing techniques perform quite differently on these three data sets with regard to the time required to build an index data structure, the number of candidate record pairs generated, and also the amount of memory required.

Despite all their limitations, the use of publicly available test data sets in data matching research has the advantage that researchers can (to some degree) compare their new algorithms and techniques to other existing algorithms and techniques. This is a much better approach for scientific progress compared to the use of proprietary or confidential data that cannot be given away, thereby making any evaluation of published research results and any comparison between techniques difficult. Ideally, research publications in data matching and deduplication that contain empirical evaluations should have been conducted on different data sets that are publicly available to illustrate how generalisable a novel data matching technique is.

[1] http://dblp.uni-trier.de/xml/

[2] http://www.imdb.com/interfaces

[3] http://www.cs.utexas.edu/users/ml/riddle/

[4] http://secondstring.sourceforge.net

[5] http://sourceforge.net/projects/febrl/

[6] http://www.hpi.uni-potsdam.de/naumann/projekte/repeatability/ datasets

[7] http://dbs.uni-leipzig.de/de/research/projects/object_ matching/fever

7.6 Synthetic Test Data

An alternative to using publicly available test data sets, which have limitations in their size and content, is to generate data that can be used to test and evaluate data matching and deduplication systems or techniques. Such synthetic data should have characteristics that are representative for the real data on which a data matching system will be applied. This means that synthetic data should contain the same or at least similar attributes, the values in these attributes should follow frequency distributions close to those in corresponding real data, and the data should also have similar error characteristics as one would expect in real data from the same domain. For example, an attribute that contains given names should contain strings that follow a frequency distribution similar to given names in a real database (such that 'Thomas' and 'Emily' occur more frequently than 'Aidyn' and 'Roberta', following current popular baby name distributions, as was discussed in Sect. 3.2) and also contain nicknames that can occur in given name attributes (like 'Bob' and 'Liz').

Generating such 'real' synthetic data can be a challenging undertaking. There are two basic approaches of how synthetic data can be created [56, 72].

- In the first approach, complete data sets are generated using (1) look-up tables that contain real attribute values and possibly their frequency distributions, and (2) rule-based techniques that generate attribute values according to rules that specify the length, distribution and content of these values. The first method is mainly suitable for attributes that contain a large number of different values, such as personal name and street and location name attributes. The second method, on the other hand, is suitable for attributes such as telephone, drivers or social security numbers which contain more structured values.

  Records consisting of a set of attributes can be generated using look-up tables and rule-sets appropriate to the content of the attributes. Various parameters have to be set by a user for example to specify the size of the database(s) to be generated and what look-up tables and rule-sets to use to generate the synthetic records. Because in real data there are commonly dependencies between attributes (for example, the given name of a person is highly dependent on their gender, while surnames depend upon the cultural background of an individual), it is of advantage if such dependencies can be modelled when data are generated. However, the more such dependencies are introduced the more complex the data generation process becomes, and the more parameter settings are required. The danger then is that a user simply leaves these parameters at their default values, rather than carefully adjusting them to their needs.

  In order to create variations of the generated records (which will then constitute the known true matching records or duplicates), variations and errors need to be introduced. Again, such errors need to follow the characteristics of real-world errors as much as possible. The conditions that govern such error imputation, such as manual keyboard data entry or optical character recognition, have been discussed in Chap. 3 in the context of data pre-processing. Essentially, the corruption of the generated records to create approximate matching records or duplicates needs to introduce variations that model the errors that occur in a real-world data entry process [72].

  Specific error parameters that a user should be able to set individually for each attribute should include: the likelihood for an attribute value to be modified in some way; the likelihood for an attribute value to be removed (i.e. set to a missing value); the likelihood to change an attribute value with a new value from the same attribute; and the likelihood to introduce character modifications such as edits (inserts, deletes, substitutions and transpositions), keyboard typing errors (replacing a character with a character neighbouring on a typical keyboard, such as 'z' and 'x'), or scanning errors (replacing a character with a similar looking character, like 'S' and '5'). For character level edits, the distribution of where modifications are applied should also be based on a parameterised model, because studies have shown that errors in real-world data do not occur randomly at any position. For manually typed names, for example, they occur mostly towards the end of names [214].

  For certain types of modifications, such as nicknames and common name variations or misspellings, having large look-up tables of known variations of values can help to generate more realistic data compared to simply inserting random modifications.

- An alternative approach to generating synthetic data from scratch, based only on look-up tables and rule-sets, is to use a real-world data set that contains records with the required content. Such data can be sourced either from a database within an organisation (such as the name and address details of all customers from a cleaned data warehouse) or from a publicly available data source (such as voters registration lists or telephone directories which are available to the public in certain countries). Such real data sets are a realistic source of variations and frequency distributions of values.

  However, because such data sets are generally well cleaned and deduplicated (one would hope especially for electoral rolls [9]), the same data corruption process described above needs to be employed to generate records that contain variations and errors that can be used in the data matching process.

Figure 7.6 shows three sets of example records that were generated with the FEBRL data set generator [56, 72]. This generator works by first creating a set of *original* records (indicated by the string 'org' in their record identifier) in the first step, followed by their modification into duplicate records (indicated by the string 'dup' and a number as there can be several duplicates generated from the same original record) in the second step. This generator allows a large number of parameters to be set so that data of different error characteristics can be generated [72].

Rec_id, age, given name, surname, street_number, address_1, address_2, state, suburb, postcode
rec-l-org, ***33***, ***madison***, solomon, ***35***, tazewell ***circuit***, trail view, ***vic***, ***beechboro***, ***2761***
rec-l-dup-0, 33, madison, solomon, 35, tazewell circ, trail view, viv, beechboro, 2761
rec-1-dup-1, 33, madison, solomon, 35, tazewell crct, trail view, vic, bechboro, 2761
rec-1-dup-2,, madison, solomon, 36, tazewell circuit, trail view, vic, beechboro, 2716
rec-1-dup-3, 33, madisoi, solomon, 35, tazewell circuit, trail view, vic, beech boro, 2761
rec-2-org, ***29***, soida, perera, ***416***, marchant place, ***weemilah***, ***nsw***, belmont, 2280
rec-2-dup-0, 29, soida, perera, 414, marchant place, wemilah, nsw, belmont, 2280
rec-2-dup-1, 92, soida, perera, 416, marchant place, weemilah, naw, belmont, 2280

rec_id, age, given_name, surname, street_number, address_1, address_2, state, suburb, postcode
rec-3-org, 29, ***jalisa***, ***wane***,25, ***prisk*** place, ***seabank***,, wa, latham, 6616
rec-3-dup-0, 29, ghialisa, wane, 25, prisk place, zeabank,, wa, latham, 6616
rec-3-dup-1, 29, jalisa, whane, 25, prisc place, seabank,, wa, latham, 6616
rec-3-dup-2, 29, jalissa, wane, 25, prisk place, seapank, , wa, latham, 6616
rec-4-org, 39, desirae, ***contreras***, 44, maltby street, ***phillip*** lodge, nsw, ***burrawang***,3172
rec-4-dup-0, 39, desirae, kontreras, 44, maltby street, phillip lodge, nsw, burrawank, 3172
rec-4-dup-1, 39, desirae, contreras, 44, maltby street, fillip lodge, nsw, buahrawang, 3172

rec_id, age, given_name, surname, street_number, address_1, address_2, state, suburb, postcode
rec-5-org, 28,***phyliss***, winter, 20, ***aspinall*** road, , qld, ***wairewa, 3887***
rec-5-dup-0, 28, phyliss, winter, 20, aspinall road, , qld, wairewa, 3881
rec-5-dup-1, 28, phyl'lss, winter, 20, aspinall road, , qld, wajrewa, 3887
rec-6-org, ***81***, ***madisyn***, sergeant, 6, ***howitt street***, creekside cottage, vic, ***nangiloc***,3494
rec-6-dup-0, 87, madisyn, sergeant, 6, howitt street, creekside cottage, vic, nanqiloc, 3494
rec-6-dup-1, 81, madisvn, sergeant, 6, hovitt street, creekside cottage, vic, nangiloc, 3494

Figure 7.6: Three examples of records created with the FEBRL data generator with different error types [72]. Original values that were modified are highlighted in bold-italics and their corresponding modified values are underlined. Two modifications were introduced into each duplicate record. The data used to generate these records consisted of name and address values taken from an Australian telephone database **a**. Typographic errors **b**. Phonetic errors **c**. OCR errors

Compared to using publicly available or proprietary and confidential data sets for testing and evaluating data matching or deduplication systems and techniques, the use of synthetic data has various advantages [72].

- Because the data have been explicitly generated, each record can be given a unique identifier and each modified record (approximate match or duplicate) that is based on a certain generated record can be given an identifier that refers back to the 'original' record it is based on. Therefore, when such data are used to test a data matching system, the match status of each candidate record pair is known and both matching quality and complexity, as discussed in Sects. 7.2 and 7.3, can be calculated. This allows the performance of data matching systems to be evaluated in detail.

- The size of the data generated and their characteristics with regard to content and variability (types and likelihoods of errors and modifications) can be fully controlled by the user. It is therefore possible to generate data that have very specific error characteristics, and to test and evaluate how well different data matching systems and techniques can handle such data. The scalability, a major challenge in data matching and deduplication, can also be tested by generating data sets of different sizes.

- The generated data sets can be published openly so that other researchers can conduct comparative evaluations on these data sets and reproduce results from other research studies. This makes research in data matching and deduplication more meaningful compared to the situation where researchers only evaluate their new algorithms and techniques on their own (possibly not published) data sets.

- The program used to generate synthetic data can be published itself, allowing other researchers and practitioners working in the field of data matching and deduplication to generate their own data that are tailored specifically to their need. For example, data specific to a country, culture or language can be easily generated by using appropriate look-up tables and parameter settings for errors and variations.

Even though synthetic data have all these advantages, the main problem with such data is still that, even with sophisticated look-up tables, attribute dependency and corruption models, such data will never be able to fully represent all the intrinsic characteristics of real-world data that make accurate and efficient data matching and deduplication such challenging problems.

Several data generators specifically aimed at generating data for data matching and deduplication have been developed. A first such generator was presented by Hernandez and Stolfo in 1995 [140]. It is known as UIS DBGen and is available from the RIDDLE repository.[8] This generator allows a user to create records and duplicates of these records using lists of names, cities, states and postcodes. It however cannot deal with frequency information for these values. This means that the frequency distribution of values will be uniform and therefore not follow the likely frequency distributions of real data. A user can set the number of records that are to be generated, the percentage and distribution of duplicates to be generated, as well as the types and amounts of errors to be introduced.

An improved generator was described by Bertolazzi et al. in 2003 [29]. It allows parameter settings that control if values in certain attributes become missing, and it also improves the variability of the created values by providing a larger number of modification and error types that can be introduced when duplicate records are generated. It is not clear if this data generator can handle frequency information, as not many details were published by the authors.

The FEBRL data matching system, described in detail in Sect. 10.2.4, includes a data generator [56, 72] that improves both the generator developed by Hernandez and Stolfo and the one developed by Bertolazzi et al. The FEBRL generator allows many parameters to be set with regard to the types, locations and likelihood of modifications applied to attribute values when duplicate records are created. Besides frequency look-up tables of attribute values, this generator also allows nickname and name variation look-up tables, as well as the specification of individual probabilities of certain types of errors, such as phonetic, keyboard and optical character recognition (OCR) errors, as Fig. 7.6 illustrates. The latest version of this generator [72] also allows dependencies between attributes to be specified (such as between a gender and a given name attribute), and it can even generate groups of records that correspond to a family. For such groups, the number of records in them, as well as their age and gender values, are drawn from specific distributions to allow realistic generation of parents and their children.

A generator similar to the one implemented in FEBRL was recently described by Talburt et al. [250]. This generator allows the creation of sequences of records that correspond to the occupancy of people as they live at different addresses over a certain period of time. The generator first creates a record that corresponds to an individual based on real data (such as publicly known real addresses). Two scenarios are possible, the first is for a single individual while the second scenario models couples living together. Sequences of records (each with a time stamp) are then generated for each individual according to the selected scenario and by introducing variations into both the name and address attribute values over time based on variations collected from real data sources.

Other data generators that can create or corrupt data in XML format have also been developed [195]. Further to the generators used in data matching and deduplication, generators for specific types of data (such as relational database tables or biological sequences) have been developed by researchers in their respective communities.

[8] Available from: http://www.cs.utexas.edu/users/ml/riddle/data.html

7.7 Practical Considerations and Research Issues

The most important consideration when evaluating the outcomes of a data matching project is if ground-truth data (gold standard) in the form of known true matches and true non-matches are available or not. If no such training data are available, then the next question is if there is a practical way to obtain or create such data that are of high quality within a reasonable amount of efforts. In some circumstances, a data matching system can be assessed using either synthetically generated data or using one of the various test data sets that have been published. For both of these the true match status of record pairs is generally known.

As was described in Sect. 7.2, the commonly used quality measure of accuracy should not be used in the context of data matching, due to the much larger number of non-matches compared to matches that are normally contained in the set of candidate record pairs. Precision and recall, as well as the f-measure, are suitable measures to assess data matching quality.

Besides the actual run-time of a data matching system on certain data sets, the number of candidate record pairs generated, or the measures of reduction ratio, pairs quality, and pairs completeness, allow the measurement of the complexity of a data matching system and its effectiveness. These three measures also allow hardware-independent comparisons of different data matching systems.

When synthetic or publicly available test data sets are used to evaluate a data matching system or technique, then it is important to be aware of the limitations of such data, and the results achieved with them should not be generalised. The performance of a data matching system or technique is dependent on the type and the characteristics of the data that are matched. Having good domain knowledge will be of high value to achieve good matching or deduplication results.

Research efforts should be aimed at developing large test collections for data matching and deduplication in a similar fashion as has been accomplished in areas of information retrieval (such as the Text REtrieval Conference (TREC) data collection[9]), or machine learning and data mining (for example the University of California Irvine (UCI) repository[10]). Such data collections should contain both synthetic and real-world data sets if feasible. Data containing personal information generally cannot be made publicly available due to privacy concerns as will be discussed in the following chapter.

An alternative to a data repository is the development of a test environment where researchers can upload their data matching algorithms. These algorithms are then evaluated on different data sets against a set of benchmark algorithms [195]. The results of such evaluations are being returned to researchers and potentially also published on a type of leader-board, indicating the performance achieved by different algorithms on various types of data matching problems.

In order to allow such a test framework to operate, an implementation-independent description of data matching algorithms is required. Similar to the predictive model markup language (PMML) initiative by the Data Mining Group,[11] an XML-based descriptive language of data matching algorithms would need to be developed.

[9] http://trec.nist.gov/

[10] http://archive.ics.uci.edu/ml/

[11] http://www.dmg.org/

7.8 Further Reading

The introductory book on duplicate detection by Naumann and Herschel [195] nicely covers the topics of quality measures, real-world and synthetic data sets, and data generators. The authors also discuss issues related to benchmarking data matching and deduplication techniques.

Christen and Goiser have provided a book chapter [71] that discusses in detail the issues involved in measuring data matching quality and complexity, and they provide an overview of a number of different quality measures. More recently, Menestrina et al. [187] discuss a novel approach on how to measure the quality of the record merging step, where records that have been classified as matches are merged into new combined records.