BUSINESS REQUIREMENTS OF A RECORD MATCHING SYSTEM

(Research Paper)
(IQ Concepts, Tools, Metrics, Measures, Models, and Methodologies)

Andrew Borthwick and Maggie Soffer
ChoiceMaker Technologies, Inc.
andrew.borthwick@choicemaker.com, maggie.soffer@choicemaker.com

Abstract: This paper seeks to describe the business requirements imposed on a record matching system along ten different dimensions. For each dimension, we present alternative requirements which different record matching clients might have. We seek to discuss the factors that might lead a client to determine that they have one requirement or another. The goal of the talk is to better prepare a client to understand their record matching needs and help them to evaluate the offerings of record matching system vendors.

Key Words: Data Quality, Record Matching

INTRODUCTION

The acquisition of a record matching system, either through purchase or in-house development, requires the business-side client to carefully think through his or her organization's requirements. There are a wide range of issues that need to be considered, which will impact the type of system that will best meet the organization's needs. This paper is intended to help clarify the thinking of record matching customers on what they are looking for to facilitate their search for the solution that is best suited to their needs.

This paper, therefore, takes somewhat the form of a list of questions that record matching consumers should use early in the requirements definition process.

BACKGROUND

This paper uses the ChoiceMaker record matching system as an example when discussing the possible scenarios for different business requirements, since the authors have the experience with it. However, this paper is relevant to issues surrounding the implementation of most record-matching systems. The following is a brief overview of ChoiceMaker Technologies' (CMT's) record matching system, ChoiceMaker 2. Further details of ChoiceMaker 2 can be found in [3].

Record matching is generally performed as a two-step process. This process is illustrated in Figure 1.

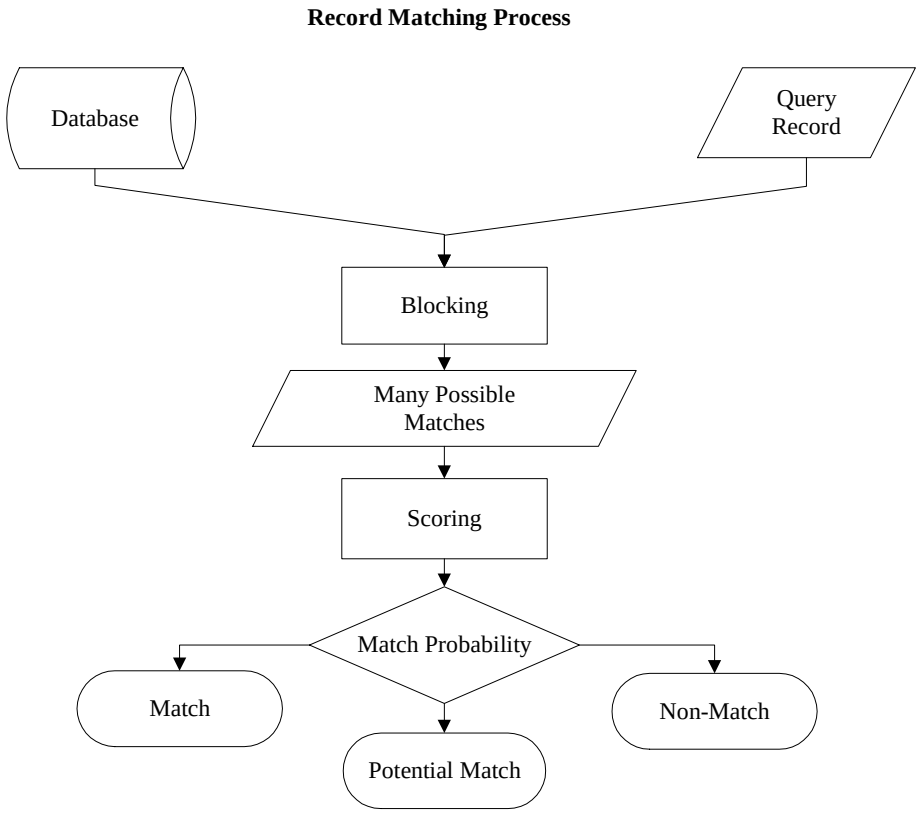

Figure 1: A two-step record matching process

Within this diagram the input to the system is called the "query record."

The Blocking stage is where the matching engine searches the target database for records that are possible matches to the query record. The objective at this stage is to retrieve all possible matches and not too many non-matches.

Scoring is where the matching engine determines for each possible match the probability that it denotes the same thing as an input record. Possible matches are categorized into matches, potential matches, and non-matches based on two user-defined thresholds. In this paper we primarily focus on the "scoring" stage.

One significant difference between ChoiceMaker and most other record matching systems is that ChoiceMaker's matching model uses weighted clues to predict a matching decision. Most other systems rely on rule-based models. ChoiceMaker's record matching models are built around a set of "clues" (commonly known in the AI literature as "features"), which indicate whether a pair of records "match" or "differ". These clues are written in ChoiceMaker's ClueMaker™ programming language [4]. Clues can be arbitrary predicates of the record-pair. Since this paper uses record matching of people as its primary examples, some sample clues for a database of people include:

- Are the first names the same and are they common, uncommon, or rare?
- Do the last names have the same phoneticization according to Soundex [8] or similar techniques?
- Is the date of birth different?

CMT's Machine Learning (ML) approach constructs a record matching model that outputs the probability that a pair of records represents the same entity. The model compares the input record with each possibly

matching record. The model is trained on a set of pairs of records that have been tagged as a "match", "differ", or "hold" (unsure). The model is trained through a machine learning technology. During an iterative model development process, the training readjusts the weights as the model is refined.

For any record matching that outputs a confidence measure for its decisions (in ChoiceMaker's case, a probability), the ultimate result relies on two thresholds, applied to each record pair, to make a final prediction. For the *match threshold*, any score above this threshold value results in a "match" decision. Similarly any score below the "*differ threshold"* results in a "differ" decision. Anything between the two thresholds, results in a "hold" decision. If the user does not want to review any automated decisions, the user can prevent hold decisions by setting both thresholds to the same value. The actions based on the example user-defined thresholds of 0.70 and 0.96 chosen by the New York City Department of Health and Mental Hygiene for their ChoiceMaker implementation are shown in **Figure 2** [9].

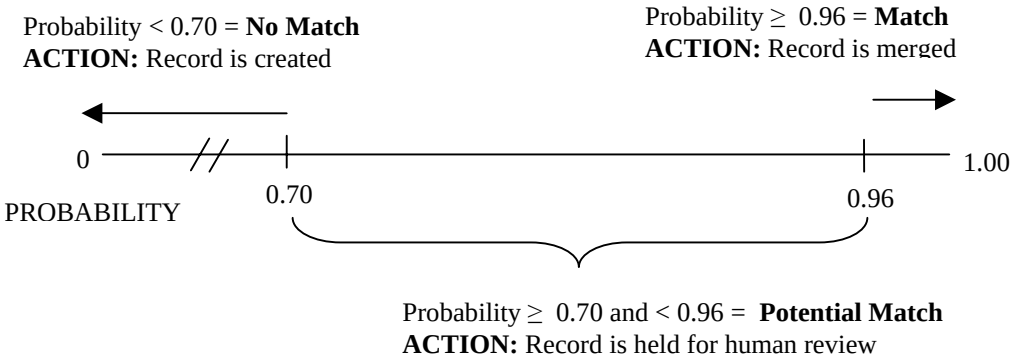

Figure 2: Probabilistic scoring of records and corresponding action[9]

RATIONALE & PURPOSE

This paper grew out of CMT's own experience with performing needs definition for our clients and sales prospects. We have found the framework in this document to be useful in helping us to determine the type of system that would best suit different organizations and the scope of the project that would be necessary to meet various requirements.

METHODS

In this paper, we lay out ten different types of questions or "dimensions" relating to different aspects of a record matching system. The dimensions were chosen based on in-house experience as well as a guide to selecting record matching systems [10]. We will discuss the types of responses that a client might give along each of these dimensions and will discuss how these different responses would yield different needs in a record matching system.

CMT gathers this information from the client throughout the life cycle of a project. At a project's kick-off, CMT uses a questionnaire to document requirements and expectations. In addition to initial requirements gathering, the development team works closely with the client team as the project progresses. Requirements and expectations change as options and issues arise during a record matching project and as the customer becomes more familiar with record matching.

Type of Task Dimension

The first question that we typically ask our client is what type of task we are being asked to solve. Here we see five broad types of tasks. In our experience, it is typical for an engagement with a client to involve handling more than one of these requirements.

One-Time Matching Project

Here the client has a specific database which needs to be deduplicated or needs to have two or more databases linked as a one-time project. There is no need to keep the database(s) deduplicated on an ongoing basis. These projects are typically done by CMT on an outsource basis, with the clients sending their database to our site. We then deduplicate or link the data and return the results to the client. These projects do not require the installation of any ChoiceMaker software on the client's site.

One frequent side-benefit of these projects is that they can serve as a relatively low-cost introduction of the record matching vendor to the client. The client can evaluate the vendor's system's accuracy on their data before paying for a license.

***Ad hoc* deduplication or linking of many different databases**

The requirements of this task are similar to the one above in that there is not a need for continuous real-time deduplication of a database. In this case, which is common in research environments, the client has a wide variety of databases which need to be deduplicated or linked with one another as opposed to one big project which needs to be done. The databases are typically relatively small, and the data will be used for statistical purposes, so very high accuracy is generally less important than the speed with which the record matching product can be adapted to a new schema. In these situations, it is very critical that the client be able to do the entire matching process using internal resources, as it would be cost-prohibitive to go to the vendor for guidance every time a new database was introduced.

In contrast to the previous two, the following three tasks all generally require a real-time solution installed on the client's site.

Maintaining the integrity of a single database

This task generally involves a two-step process directed toward the goal of maintaining a single deduplicated database. Step One is to deduplicate an existing database. In Step Two, the database must be kept free of duplicates while records are added. For Step Two, the system is generally architected such that before any record *r* is added to a database *D*, the client system calls the matching server to determine if *r* is already in *d*. If the matching system returns a matching record *s*, then *r* is linked to *s*. Otherwise, *r* is added to *D*.

Note that these two steps are closely related, because deduplicating an existing database can be implemented by loading every record in the existing "dirty" database to a new "clean" database, one record at a time. Hence in ChoiceMaker's case, these two steps are implemented with essentially the same architecture, and the same record matching engine determines whether each record is already present in the database.

Linking multiple databases via a master index

In general, the best approach to linking multiple databases is to create a new master database which establishes a unique ID for every individual in any of the organization's databases, and which aggregates the information necessary for matching into the master database. Every record in each of the satellite databases is then associated with the corresponding record in the master database via a master ID.

This task is very similar to the previous task in that the master database is loaded with data, and the records in the satellite databases are assigned a master ID in Step One, and then in the ongoing Step Two the integrity of the master databases and the satellite databases is maintained by ensuring that no record is added to any database without first checking whether it is present in the master database.

Doing an approximate search on a database

Here the client desires to be able to do a real-time approximate search of a database. For instance, the client might type in "Andy Borthwich" as the name of a person being sought and retrieve a record for "Andrew Borthwick" from the database. This task is very similar to the previous two, but there are two key distinctions:

1. The user generally types less information into an online query form than would be present on a record coming into the system through another means. For instance, even if records in the database generally have identifying information other than first and last name, the user might only type those fields when making an online query
2. The client generally has a much higher tolerance for false matches in this scenario, because all of the matches returned are just going to be displayed to the user

This task requires that the record matching system provide a real-time interface (see below). It also requires an interface tuned to these requirements. In general, the record matching system should be tuned to return a greater number of possible matches in this case.

Note that a variation of this scenario is when a user is manually keying a new individual into the database. This case is similar to the previous two tasks in that the query record is relatively complete, but it is more like the search scenario in that the results returned by the record matching system are always human-reviewed.

Type of Engagement Dimension

The next question is whether the client wants to work directly with the record matching vendor on the project or indirectly through a prime contractor. CMT has found that the type of engagement is heavily dependent on project scope. When record matching is just one component of a larger systems integration project, such as the construction of a master client index across a public health registry [9], for instance, CMT generally works as a subcontractor. On the other hand, when record matching is the only thing required for the project (usually because record matching is being added to an existing database), we typically have a direct relationship with the client.

Note that a prime contractor can insulate the client from needing to know some details of the record matching system (the exact interfaces, for instance), but not others. For instance, it will generally be up to the client rather than the prime contractor to do at least one audit of the record matching system's accuracy to determine whether it is matching records as the client intended.

Required Functionality Dimension

The third question we typically ask a client is the exact scope of the organization's needs. These needs always include a record matching solution, but might also include a need for standardization and for linking and merging support.

A pure solution would have roughly the following API:

- Input: Demographic information about a person (or about a corporation or whatever type of data is being stored in the database)
- Output: A set of ID's in the database that match the input

Note that the output will often contain some additional information about the system's confidence in the match. For instance, for each ID, ChoiceMaker returns both a decision ("match" or "possible match/hold for human review") and the probability of match that ChoiceMaker assigns to the pair. For instance, ChoiceMaker would return something like the following:

Record ID	Decision	Probability
42	match	95%
57	hold	46%

Table 1: A pure record matching system returns record id's that matches the input record.

Hence, Record 42 matches the input with a probability of 95%, so ChoiceMaker declares it a "match". Record 57 matches the input with a probability of only 46%, so ChoiceMaker determines that it should be held for human review.

This is a very simple interface, which is appropriate if the client only needs matching. In the ChoiceMaker system this is offered as the basic interface for all customers. For more complex record matching needs, some of the following outputs might be required:

Standardization
Output standardized values for the fields in the input. For instance, one parses the name into first, middle, and last names. Addresses are parsed into city, state, and zip code. Abbreviations might be standardized, addresses might be placed in U.S. post office-certified CASS format, etc.

Support for linking and merging operations
If a record matching job also encompasses updating a database based on the output of the record matcher (which is usually the case), clients often find that they need to address complex issues of linking or merging large sets of matching record ID's. This can sometimes involve a need to grapple with complicated issues of "transitive matches".

If the record matching system determines that Record A matches B, B matches C, and C matches D, then we say that these records all belong to the same "equivalence class" [6], because according to the record matching system they all match (they all represent the same person, so they are "equivalent"). However, we have found that there are a significant number of cases where our customers want to override the equivalence classes produced by a simple pair-wise comparison of records. For instance, if the record matching system determines that Record A matches B and B matches C, but does not see a match between A and C, some clients may want to link the three records, some might not want to link, and some might want to send the records to human review. In this case, other clients might want to link them depending on criteria like the degree of confidence that the record matching system has in each match. The decision-making criteria can grow quite intricate when one considers that a single record may be a clear match to one record and a "hold" relative to another.

A post-processing module from the vendor of a record matching system can be very helpful in allowing the client to specify how these issues should be resolved. A post-processing module can also be helpful in allowing the client to identify the equivalence classes in the first place. The algorithms for identifying an equivalence class of records from a set of pairwise relations are well known (by representing the pairwise relationships as a graph and then performing a "depth-first search" on the graph [5]), but are not trivial to implement.

Type of Interface Dimension

At a high level, one can describe interfaces according to three parameters:

1. What are the semantics of the interface? In other words, what information is being passed back and forth? This was discussed earlier in the "Required Functionality" dimension and won't be repeated here.
2. Is there a real-time or asynchronous interface?
3. What is the technical architecture of the interface?

We will consider parameters 2 and 3 below.

Real time or Asynchronous Interfaces?

An asynchronous (or "batch") interface is appropriate for one-time deduplication of databases or for processing large numbers of records when the client does not need a real-time response for each record. In general (at least with ChoiceMaker's architecture), it is possible to process a large batch of records more quickly when they are submitted to the record matching system as a group rather than one at a time. A batch interface is clearly a necessity when the client's database is not a relational database (when it is an Excel spreadsheet, for instance). A real-time interface is clearly required when, for instance, a clerk is performing an approximate database search or is doing data entry.

The technical architecture of the interface

A real-time interface could be implemented in many different ways. Some common examples include a J2EE (Enterprise Java Bean) interface, a web service (SOAP) interface, or via CORBA, COM/DCOM, or as a native library for a language such as C or Java. Most record matching vendors offer multiple interfaces in order to satisfy a range of clients.

A batch interface could be implemented with any of the above interfaces or it could be a simple file-based interface where the file might be, for instance, XML, a database table, or a simple file of comma separated values. Again, most major record matching vendors support a variety of interfaces.

Data Type Dimension

The next question to consider is what the records in the database represent. This document uses person matching as an example, which is a very common matching problem. Person matching can be done either at the person or at the household level. However, there are many other types of data that could require matching.

Matching of corporations is another common matching problem and presents a number of distinct challenges. Corporate names can be challenging to match because of the number of different ways in which they can be represented. For instance, one must match "International Business Machines" with "IBM" and "Intl Bus Mach". Furthermore, there is a crucial question of the definition of a match. Should "General Motors" be matched with "General Motors, Canada"? Should "Disney" be matched with its subsidiary, "ABC, Inc."?

Clearly, there are many other types of data that might need to be matched. Some examples include documents, spare parts in a catalog, and financial securities.

Data Quality Dimension – The difficulty of the matching problem

Record matching problems differ widely in terms of complexity, but the complexity of the matching problem can be broken down into three components: the cleanliness of the data, the number of fields available for matching, and the complexity of the business rules required to match the fields.

Data Cleanliness

Data containing more errors is typically more difficult to match, because with noisier data there is a greater chance that two fields will match by chance or that two identical fields will be transcribed differently.

The cleanest data typically comes from data that is being actively used to drive business transactions. For instance, customer billing data is typically kept accurate, at least for the address, or else the bills would not be delivered to the appropriate party. Other data tends to be accurate, because it is typically filled out with great care. For instance, parents are typically very careful in filling out vital records data (birth certificates, for instance), because they are aware that this birth certificate will be a vital document for their child throughout his or her life.

By contrast, some sources tend to be very dirty. Forms filled out on the Internet, for instance, are notoriously dirty, because they tend to be filled in hurriedly and people might, at times, want to disguise their identity to avoid "spam" email. Forms filled in by the general public by hand are also often problematic, as either optical character recognition (OCR) or manual transcription can introduce errors in interpreting the handwriting. Note though, that OCR tends to produce errors with the substitution of letters that are visually confusable ("o" vs. "a", for instance). Manual transcription suffers from this to a somewhat lesser extent, but adds in the problems of typographical errors (for instance hitting "b" rather than "v" on the keyboard). Homonyms and the confusion of rare names with common names (e.g., "Ashlee" with "Ashley") become a problem when information is being dictated to a clerical worker.

We should also consider, especially when matching multiple data sources, whether the data was gathered in different time periods. For instance, if one is matching a database of kindergarteners against a vital records database, the five years that have passed between birth and school entry are likely to produce many inconsistencies such as address changes and names changing due to nicknaming or due to a mother marrying and changing her children's names along with her own.

Proper fielding of data is another issue. Is the name parsed into first, middle, and last name, or is the entire name frequently placed in the "last name" field?

Number of fields available for matching

Databases whose schemas contain many different matching fields should be easier to match than databases with few fields. For instance there is much more ambiguity in a database which only contains first name, last name, and birthday than in one which also contains fields such as mother's maiden name, mother's date of birth, place of birth, etc. A record matching system should be able to leverage this extra information, but it can only do so if the system is sufficiently flexible to allow the record matching model to incorporate the extra information when it is available.

At CMT, we have found that we very frequently have novel information available for record matching. For instance, for a project to deduplicate and match a database of children's immunizations with a database of children's blood lead tests [9], ChoiceMaker was able to exploit such fields as the date on which a medical event was performed, the physicians name, and a variety of medical record identifying numbers. For a project involving matching children in a statewide K–12 student database, we were able to exploit education-specific fields like building code, district code, expected date of graduation, etc.

Finally, there is the question of how many of the fields that are supposedly available for matching are actually available on average. Are fields frequently filled in with 'generic' values, like values of "Boy" or "Unknown" for "First Name"?

Business Rule Complexity

For some databases, the only thing required to make an accurate decision as to whether two records match is to do a number of fairly standard comparisons on all the different fields that comprise the two records. We have found, however, that there are many cases where sophisticated logic is required to make an accurate decision. We will cite three examples:

- When matching a nationwide database of names, addresses, and birthdays, we built a record matching clue which looked for "snowbirds", retirees who lived in the Northeast or Midwest in the summer and in Florida or Arizona in the winter. This clue had a number of conditions: one address had to be in Florida or Arizona and one in the Northeast or Midwest. The age had to be over 60, etc.
- Twins are particularly difficult to identify in an immunization registry because they share the same last name, address, birthday, parents, etc. One good indicator, however, is that hospitals will often give them sequential medical record numbers when they are born (because they are entered into the database one after another). Hence, we wrote a clue which checked whether the medical record numbers differed by 1 in situations where there was a possibility that a pair of children might be twins.
- One medical organization supplying data to the New York Citywide Immunization Registry always submitted their data with the birthdays occurring on the first of the month because, apparently, the system only stored the child's year and month of birth. Hence we had to write special logic into a special "birthday match" clue which increased the likelihood of a match when the days of birth differed and one of the records was from this organization.

Accuracy Dimension

Measurements of accuracy

When talking to clients about their accuracy needs, the client's usual first response is "the more accuracy the better". However, this is an answer that usually needs to be refined as there are different measurements of record matching accuracy and achieving the highest levels of accuracy often involves making certain tradeoffs.

A record matching system's accuracy can be described in terms of three parameters: the percentage of "false positives", the percentage of "false negatives", and the percentage of records on which the system makes no decision. Let us consider each of these cases in turn.

False positives or "false matches" are cases where the system identifies two records as representing the same individual which in fact represent different people. In a children's immunization registry [9], this is a serious error because a physician might look at the database, see a vaccination from one child on the record of another child, and conclude that the child has received a vaccination that he/she did not in fact get, and not administer a needed vaccine. On the other hand, when conducting a broad population survey such as [11], this is not very important if the survey is only intended to sample the population and not provide complete coverage. One person will simply be omitted from the sample, which should not significantly skew the data.

False negatives or "false differs" are cases where the system fails to identify two records as representing the same individual which do co-represent. One area where this type of error is very serious is in law enforcement and anti-terrorism situations. For instance, when matching a list of airline passengers to a list of suspected terrorists, failing to identify a match might result in a hijacking, while falsely making a match only results in the passenger and the passenger's baggage being carefully searched. Although a large number of false positive errors in terrorism situations can raise civil liberties concerns [7], clearly the bias is towards accepting more false matches so as to avoid false differs. In an immunization registry, by contrast, a false differ merely results in a duplicate record in the registry, which means that some

vaccinations will be recorded on one record and others on another record. The medical consequence here is that the physician may be led to believe that a child did not receive certain vaccinations which the child did receive, and administer a second, unnecessary vaccination. Repeating a vaccination is usually safe[1].

The third accuracy parameter is the percentage of records on which the system makes no decision. These records are generally held for human adjudication as to whether they are matches or not. We will refer to these as the "hold" records. This parameter arises because a client could place one of three requirements on accuracy:

1. Limit the false match error rate to X%. No limit on the false differ error rate.
2. Limit the false differ error rate to Y%. No limit on the false match error rate.
3. Limit the false match error rate to X% AND limit the false differ error rate to Y%. All records on which the record matching system is not sufficiently confident should be held for human review.

In case 3, the record matching system vendor seeks to achieve a tradeoff between accuracy and human review workload which is as painless as possible for the client. However, even if one hypothesized an ideal record matching system which perfectly mimicked the decisions that a client's expert record matcher would make on every case, for some pairs of records it is inherently impossible for even a human expert to determine whether the records represent the same individual. This could be due to the records being either incomplete or containing conflicting data. Table 2 contains examples of incomplete record pairs and record pairs that are unclear due to conflicting information.

Record 1	**Record 2**	**Notes**
Name: John Smith SSN: Address: 477 Cedar Street	Name: John Smith SSN: 123-45-6789 Address:	Data is incomplete, due to a common full name and no additional matching information
Brendan Hughes Address: 564 Hickory Pl.	Brenda Hughes Address: 564 Hickory Pl.	Data is incomplete, could indicate twins or a typo, need additional information to confirm
Name: Jean Smith Phone #: (337) 555-6676	Name: Gene Smith Phone #: (337) 555-5676	Data is conflicting due to different spelling of names and phone #s which could be a typo or could be different people
Name: Alice Jones SSN: 123-45-6789	Names: Lois Avon SSN: 123-45-6789	Data is conflicting, same SSN could be a typo or names could indicate questionable identity issues

Table 2: Record pairs which may or may not be "matches"

Consequently, it can be seen that statements such as "our system is 99.5% accurate" are somewhat meaningless in the absence of information about how many records are marked as "hold". CMT likes to judge the quality of its systems by measuring the percentage of records that need to be human reviewed to achieve a given level of accuracy [2]. This methodology is valid for any system (like ChoiceMaker) which outputs a degree of confidence along with its decision. For these systems, we can run a test on a set of record pairs which have been marked with the correct matching decisions. We first raise the threshold dividing the "match" records from the "hold" records (see Figure 2) until the error rate on the matches is below the false match requirement. Similarly, we lower the bottom threshold until the error rate on the record pairs marked "differ" is below the false differ requirement. The record-pairs between these two thresholds need to be human reviewed. We can then make statements like "We have tested our system as requiring clients to review only 5% of the record pairs coming into the system while achieving 99.5% accuracy with respect to both false positives and false negatives."

Once the record matching system vendor has completed tuning the record matching model for the client's database, the client will be faced with a "hold vs. accuracy" graph like Figure 3. This graph shows, for instance, that to achieve 99% accuracy, the client needs to review about 4% of the record-pairs. The client then needs to determine how much accuracy they need and how many records they need to review. One question to ask to help clarify the issue is to rephrase your accuracy requirements in terms of the amount of work required. For instance, a 99.9% accuracy requirement means "I am willing to review a thousand pairs of records which are true matches in order to find one record which is a false match."

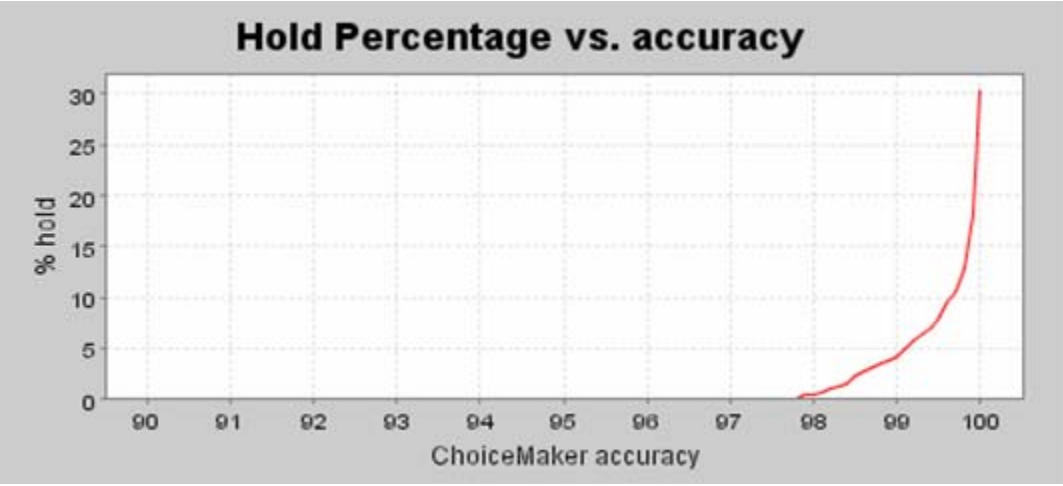

Figure 3: Hold Percentage vs. accuracy

Platform Dimension

Most organizations will have a database server (or farm) and an application server (or farm) and will require the record matching software to be compatible with both servers. Record matching vendors typically support Unix, Linux, and Windows OS. Some vendors will also support operating systems used by mainframes or AS400s.

Next you need to identify the databases used by the source files and the target files. Many vendors will support Oracle, MS SQL Server; and other database applications such DB2, MySQL or Sybase; or delimited text files. In addition, many vendors will support proprietary applications such as Siebel, PeopleSoft, and Informatica, via adapters.

Hardware requirements will vary based on your performance requirements and the size of your database. However, fast servers are generally required for record matching systems. Some suggested specifications for a moderate size database of 5 million records are at least 4 GB RAM, 15,000 RPM SCSI hard drives, and two to four 3 GHz CPU's.

Performance Dimension

When evaluating the performance dimension of a record matching project, the client needs to know how many records they want to have processed in a given amount of time. Can a job run eight hours overnight, or will they need a real-time response for each record processed?

Real-time processing of records is typically used for data entry, user queries, or other instances when a real-time response is required. A ChoiceMaker benchmark for real-time matching, for instance, is under 0.5 seconds per record. This speed is generally acceptable for supplying a user with real-time response, but it is not efficient if you need to deduplicate a ten million-record database.

Batch records are used to process a large number of records at one time. Batch processing may be performed as a scheduled job, or it may be run ad hoc as files of records are added to the database. Batch processing is typically much faster. ChoiceMaker, for instance, has been tested as processing over 800,000 records per hour on a single-processor CPU with moderately complex data types. Note, however, that due to the initial overhead in batch processing, our real-time algorithm works better when adding a small number of records (< 1,000) to an existing database.

Client Involvement Dimension

How involved does the client want to be?

Based on the client's experience and business goals, a client will choose to have different levels of involvement in the development of their record matching system.

Companies that have ongoing record matching requirements may find it is cost-effective to have staff resources available to work closely with the record matching vendor. These companies are more likely to be involved in refining project requirements, defining acceptance criteria, and evaluating the results. It can be advantageous for these companies to spend the time necessary to gain in-house expertise in record matching methods and achieve a good understanding of their vendor's system.

On the other hand, it is probably not cost-effective for companies with a one-time record matching project to gain record matching expertise. These companies will typically look to the vendor for assistance in defining requirements and evaluating the results.

Client involvement during the development phase

Regardless of the overall level of involvement, there are some areas in which a client must work with the vendor to ensure that the matching system meets their requirements. This is most true in the development process. Ultimately, it will be the client who determines whether a pair of records should be marked as a "match" or not. Only a person familiar with the data can properly evaluate the results of the record matching model and determine its accuracy.

CMT leverages a client's knowledge of their data during two phases of the development process, the machine learning phase and the review phase. CMT trains and reviews a record matching model using record pairs that have been marked "match", "differ", or "hold" by a human. We request that the client mark some of these record pairs. This way, we are sure that the model is based on the client's business rules and not our own assumptions. During the machine learning phase, we use half the data marked by the client to train the model. After a model is complete, the client is asked to review its quality by running it on the second half of the marked record pairs, which CMT has never seen. This ensures a controlled test of ChoiceMaker's accuracy.

Client involvement with acceptance criteria

The client must also work with the vendor to come up with a set of acceptance criteria based on the Data Quality and Accuracy Dimensions. Since this set of criteria is not based on hard figures, the vendor should interview the client and understand their needs and expectations.

Examples of questions that CMT asks to gather accuracy criteria are:

- Does the client want to perform human review? If, so, what kind of tradeoff can be made between accuracy and human review?
- Is the client more concerned about false matches or false differs?
- What are the criteria used by a human reviewer to make a matching decision?
- Are there any special cases that should always result in a "match" or "differ" decision?
- Are there known anomalies in the data that should be represented by special logic in the record matching model?

The answers gathered from this interview will be used to determine when the model is complete. It also provides essential information to help the vendor in the development of the record matching model.

Other areas of client involvement

Many clients want to treat the record-matching model as a black box. They don't necessarily want to know how the model arrives at its decision, but they do want to be able to evaluate its results. These clients tend to be interested in looking at the final decisions made and the statistics associated with the model. This level of involvement does not require training – a user's guide is generally sufficient for a client to perform these tasks. CMT and most other vendors offer professional services to create the model and perform the associated tasks that this type of client is not interested in performing.

Other clients may want to understand how the entire record matching system works. In addition to reviewing decisions and metrics, they may want to tweak their model so it will work on other sets of data, or even create a new model for another project. This requires training the client on all aspects of the vendor's system. The training course should also discuss the issues surrounding the creation of a good record matching model. CMT and most other vendors offer training courses for this level of involvement.

During the integration process

The final area of client involvement is the integration of the matching model output into the client's production system. Some clients might hire the vendor or a Systems Integrator to perform this task. However, the client and the record matching vendor must be involved in this process to understand the requirements for successful completion of the client's project.

RESULTS

For obvious reasons, we are not able to comment on the usefulness of this framework for use by a client in selecting a record matching system. For CMT, however, this framework has proved helpful in guiding the development of our record matching system. A major goal of CMT's development efforts, as funded by a grant under the National Science Foundation's Small Business Innovation Research Program has been to turn the original prototype ChoiceMaker 1 system into a system capable of working across as many different record matching dimensions as possible. The need to handle a wide variety of different data types, levels of quality, and accuracy requirements was a primary motivation behind our efforts in areas such as machine learning, the ClueMaker Programming Language, and our work to architect the system in such a way that our core record matching logic would work across a variety of interfaces (EJB, web service, file-based, etc.), DBMS's, and operating systems.

DISCUSSION

Clients may find it useful to think about their record matching needs along these 10 different dimensions. These 10 dimensions can serve as a checklist determining the features that a client needs to ask about when shopping for a record matching system.

As noted above, for the record matching system vendor, the variety of responses that are possible along these 10 dimensions present a challenge in designing the record matching system. In general, one wants to build a single system which can be configured to handle as many different combinations of responses as possible.

In addition to traditional requirements documents and change control management procedures, CMT exploits its iterative development process as an information gathering tool. During the development process, CMT presents record matching model releases to the client for review. We have found that as the client reviews and tests these model releases, they will often identify new or different requirements that were not considered previously. This method of information-gathering has proven to be very effective in keeping the client involved with the project and ensures that there is no miscommunication as to the direction the project is taking.

LIMITATIONS

Given that record matching is a fairly obscure discipline, it is often difficult for a client to acquire the necessary expertise to properly address all of these record matching dimensions. We have found that most of our clients require significant support from project managers, sales people, and technical staff to fully flesh out all of these issues. Ideally, customers would come to us knowing all of their requirements, but we have found that it is part of our job to flesh out some of these issues.

CONCLUSION

This paper has described 10 major ways in which record matching systems and customer needs vary. Clients should think through their requirements in each of these areas when selecting a record matching system vendor. Record matching vendors should seek to position their product to respond to a wide variety of possible customer requirements along each of these dimensions of need.

REFERENCES

[1] Atkinson, W. L., Pickering, L. K., Schwartz, B., Weniger, B. G., Iskander, J. K., and Watson, J. C., "General Recommendations on Immunization," *Morbidity and Mortality Weekly Report.* Feb 2002.

[2] Borthwick, A., Measuring Record Linkage and Deduplication Accuracy. *Enterprise Data Forum*, November, 2003, Philadelphia, PA.

[3] Borthwick, A., Buechi, M., and Goldberg, A. "Key Concepts in the ChoiceMaker 2 Record Matching System". First Workshop on Data Cleaning, Record Linkage, and Object Consolidation. July 2003, Washington, D.C.

[4] Buechi, M., Borthwick, A., Winkel, A., and Goldberg, A., "ClueMaker: A Language for Approximate Record Matching," *Eighth International Conference on Information Quality*, Cambridge, Massachusetts, Aug. 2003.

[5] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. Depth-first search. In: *Introduction to Algorithms*, eds. Cormen, T. H., Leiserson, C. E., and Rivest, R. L. Massachusetts: Massachusetts Institute of Technology, 1990.pp. 477-485.

[6] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. *Introduction to Algorithms*, Massachusetts, USA: Massachusetts Institute of Technology, 1990.

[7] Davis, A., Why a 'No Fly List' Aimed at Terrorists Often Delays Others *The Wall Street Journal*, pp. A14. Apr 22, 2003.

[8] Newcombe, H. B. *Handbook of Record Linkage: Methods for Health and Statistical Studies, Administration, and Business*, New York: Oxford University Press,

[9] Papadouka, V., Schaeffer, P., Metroka, A., Borthwick, A., Tehranifar, P., Leighton, J., Aponte, A., Liao, R., Ternier, A., Friedman, S., and Arzt, N., Integrating the New York Citywide Immunization Registry and the Childhood Blood Lead Registry *Journal of Public Health Management and Practice*, to appear in Nov, 2004.

[10] Raab Associates, "Raab Associates Guide to Customer Matching Systems," Chappaqua, NY, 2004. Available for purchase from Raab Associates at http://www.raabassociates.com/database/software.htm.

[11] World Trade Center Health Registry. WTC Health Registry Data Snapshot: Quarterly Enrollment Update (July 2004). Available at http://www.nyc.gov/html/doh/pdf/wtc/wtc-report2004-0803.pdf.