Hierarchical Cluster filtering Processing for Large Scale Web Search Engine

P Chandrashekar Reddy [1], Y Sushma[2], B Aruna[3]
[1]*Asst. Professor, Dept of CSE, Malla Reddy Engg. College,AP,INDIA,*
[2] *M.Tech Student, Malla Reddy College of Engg.&Technology,AP,INDIA,*
[3] *M.Tech Student, Malla Reddy College of Engg.&Technology,AP,INDIA,*

Abstract— *Data cleaning and integration is typically the most expensive step in the KDD process. A key part, known as record linkage or de-duplication, is identifying which records in a database refer to the same entities. This problem is traditionally solved separately for each candidate record pair. We propose to use instead a multi-relational approach, performing simultaneous inference for all candidate pairs, and allowing information to propagate from one candidate match to another via the attributes they have in common. Our formulation is based on conditional random fields, and allows an optimal solution to be found in polynomial time using a graph cut algorithm. Parameters are learned using a voted perceptron algorithm. Experiments on real and synthetic databases show that multi-relational record linkage outperforms the standard approach.*

Keywords— *De-duplication, Hierarchical clustering, web search engine, Data cleaning and data integration.*

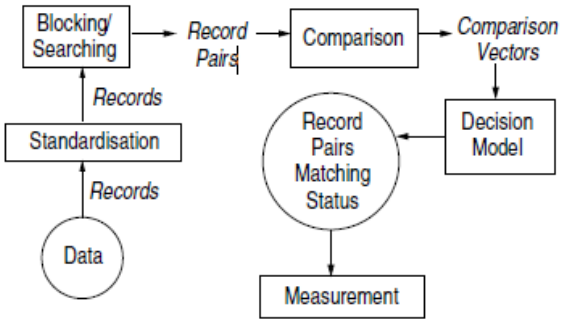
I . INTRODUCTION

Record linkage, the problem of determining when two records refer to the same entity, has applications for both data cleaning (deduplication) and for integrating data from multiple sources. Traditional approaches use a similarity measure that compares tuples' attribute values; tuples with similarity scores above a certain threshold are declared to be matches. While this method can perform quite well in many domains, particularly domains where there is not a large amount of noise in the data, in some domains looking only at tuple values is not enough. By also examining the context of the tuple, i.e. the other tuples to which it is linked, we can come up with a more accurate linkage decision. But this additional accuracy comes at a price. In order to correctly find all duplicates, we may need to make multiple passes over the data; as linkages are discovered, they may in turn allow us to discover additional linkages. We present results that illustrate the power and feasibility of making use of join information when comparing records.

Record linkage is the problem of identifying multiple records that refer to the same real-world entity. In genealogical databases, it is the problem of identifying when individuals situated in different pedigrees refer to the same real-world individual. Being able to link records in genealogical databases has value to people engaged in genealogical research because it condenses search results and helps people identify when their work overlaps with the research of others. It also has value to medical researchers trying to understand the hereditary nature of cancer, heart disease, and other illnesses. Unlike most record linkage problems, record linkage in genealogical databases usually allows one to utilize a broad range of features, since records are often situated in the context of pedigrees. For many individuals within a pedigree, dates and locations of birth, marriage, and death are usually available, as well as information about children, spouses, siblings, and parents. Often individuals within a pedigree are

identified through different vital records by different genealogical researchers. Furthermore, the linkage problem can be cast as a graph-matching problem, since the decision to link (or not to link) two individuals influences the decision to link individuals related to them [5]. Finally, any record linkage problem has the issue of determining the similarity of different names [3,6], determining for example the probability that Peg, Peggy, and Margaret name the same person. Linked genealogical databases can provide insight into name similarity since we can identify the various names associated with linked records. Since the linking of genealogical databases should be quite accurate due to the broad range of features and the ability to use graph-matching concepts, one could be reasonably confident in the names that are found to be similar.

2. ***Record Linkage and Record Matching Architecture***



Record linkage is the task of identifying records corresponding to the same entity from one or more data sources. Entities of interest include individuals, companies, geographic regions, families, or households.
Record linkage has applications in customer systems for marketing, relationship management, fraud detection and government administration. These applications can be classed as 'administrative', because record linkage is used to make decisions and take actions regarding an individual entity. In many data mining projects it is necessary to collate information about an entity from more than one data source. If a unique identifier or key of the entity of interest is available in all of the data sources to be linked, conventional 'join' operations can be used for record linkage, which assumes error-free identifying fields and links records that exactly match on these identifying fields. However, real-world data is 'dirty' and sources of variation in identifying fields include lack of a uniform representation or format, misspellings, abbreviations, and typographical errors. Record linkage can be considered as part of the data cleaning process, which is a crucial first step in the knowledge discovery process [4]. Fellegi and Sunter [5] were the first to introduce a formal mathematical foundation for record linkage, following a number of experimental
papers that were published in the medical domain since 1959 [11]. Winkler [12] extended and enhanced the original model.
No matter what technique is used, a number of issues need to be addressed when linking data. Figure 1 shows the information flow diagram of a typical record linkage system as implemented in TAILOR [3] and Febrl [2]. Often, data is recorded or captured in various formats, and data fields may be missing or contain errors. Standardization is an essential first step in every linkage process to clean and standardize the data. Since potentially every record in one dataset has to be compared with every record in a second data set (i.e. the number of record pairs to be compared grows quadratic ally with the number of records to be matched), blocking or searching techniques are often used to reduce the number of comparisons. In this paper, we focus on the blocking/searching component in the information flow diagram. The performance
bottleneck in a record linkage system is usually the detailed comparison of record pairs. A good blocking method can greatly reduce the number of record pair comparisons and achieve significant performance speed-ups. The main contribution of this paper is the development of a fast adaptive filtering algorithm, which can be combined with any blocking method as a post-processing step to further reduce the number of record pairs for comparison with minimal accuracy loss. A key innovation is that the filtering is applied only to blocks with a significant number of records.

3. Adaptive Filtering:

The previous section shows that the number of record pairs generated by any blocking method depends on the number of blocks it generates (linearly) and their sizes (quadratically). Very large blocks have therefore dominant effects on the efficiency of blocking methods. It is generally difficult to avoid large blocks no matter what blocking methods/keys are chosen. For example, the block 'blac' will be much larger than the block 'szep'if the first 4 characters of the variable surname are used
as the blocking key, since frequent surnames such as 'black', 'blackburn' and 'blackman' will be included in the block 'blac'. To improve the blocking efficiency, we propose an adaptive filtering algorithm as a post-processing step of the blocking process. The filtering is adaptive in the sense that the number of blocks to be filtered is dependent on the results from a blocking method. Filtering is only applied to larger blocks to filter out potentially unlikable record pairs.
Specifically, it is observed that not all record pairs within the two similar blocks are potential matches, as the blocking key used might be incomplete or contain errors, or the blocking key does not have enough discriminating power. As a result, complete variables of blocking key or other information (mainly name and address) in records can be used to perform fast approximate comparisons to filter out unlikable record pairs before detailed comparisons are performed. The key objective of filtering is therefore to efficiently eliminate those unlinkable record pairs using the information contained in a chosen filtering variable. A filtering variable should be chosen to be different from the blocking key in the following sense. It should contain independent information differentiated from that of the blocking key
for the filtering process to be able to quickly remove unlikable record pairs. For example, a filtering variable can be an entire surname instead of the first 4 characters of the surname, or it can be a string composed of the complete given name and surname. Next, we need an efficient method for deciding whether a record pair is unlikable. We use the fast approximate comparison method proposed by Gravano et al. [6] for our adaptive filtering. The method was initially devised for performing efficient approximate string joins by exploiting q-gram properties. One of the measures used for string comparison is the edit distance. Gravano et al. [6] relate simple q-gram properties to lower bounds on string edit distances. Given a filtering variable (a string), we convert it into a list of bigrams.
In the following two subsections, we describe some key properties of bigrams and show how they can be used to perform fast filtering without calculating the edit distance.

3.1 Length Filtering

It is observed that string length provides useful information to quickly eliminate those very dissimilar strings. Dissimilar strings are defined to be those that are not within the desired edit distance. Specifically, it can be proved [6] that if two strings s1 and s2 are within an edit distance k, their lengths cannot differ by more than k. For our particular case, if the string length difference
of the filtering variable values in two records is larger than a predefined value k, this pair of records is declared as unlinkable and eliminated from the record pair list.

3.2 Count Filtering

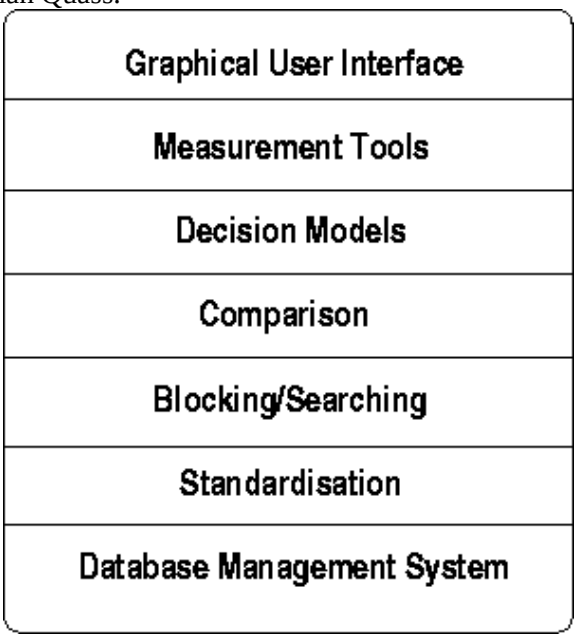
The above length filtering is not very effective for strings with a uniform length distribution. There are other features that can be used to perform the filtering. One such feature is based on the observation that similar strings share a large number of common bigrams. The basic idea of count filtering is therefore to make use of the information conveyed in the sets Bs1 and Bs2 of bigrams of the strings s1 and s2, ignoring the positional information of the bigrams, in determining whether s1 and s2 are within the edit distance k.

3.3 Creating Labeled datasets

The Church recently embarked upon an effort to combine the IGI, AF, and PRF databases into a single database. We plan to link records initially in batch mode as we populate the database from

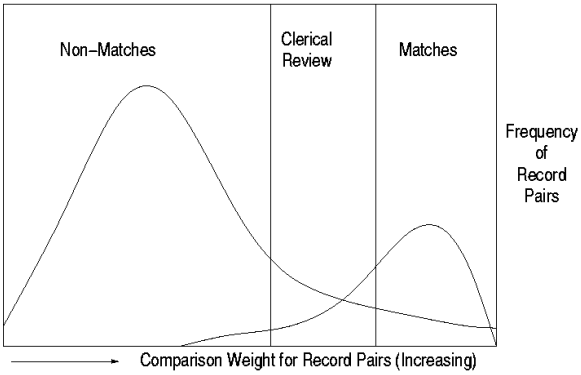
these sources, and then interactively as new pedigrees are submitted. Linking records within the IGI is similar to a traditional record-linkage problem, since little relationship information is included. A difference is that we hope to reconstruct multi-generation pedigree structures from the data in IGI that was originally submitted in pedigree format. Linking records within AF and PRF and between these two databases and the IGI records that have been placed into pedigrees allows for the broad range of features and graph-matching opportunities discussed in Section 1. As a first step in linking the records, we plan to create sample datasets from each of the three sources and to manually label linked records within and between the samples. We again intend to use a Fellegi-Sunter-based statistical linking algorithm. We hope to augment the algorithm with name-similarity and location-proximity metrics. In addition, we plan to use features based upon family relationships and hope to integrate graphbased linking techniques into the statistical algorithm.

A basic statistical record-linkage algorithm will be run over the datasets. Standard blocking techniques will be used to reduce the number of pairs of individuals to consider. Once the algorithm has been run, random sampling will be used to determine the lower-confidence threshold, below which all pairs will be set automatically to .not linked.. Likewise, sampling will be used to determine the upper-confidence threshold, above which all pairs will be set automatically to .linked.. Pairs whose linking score falls between the lower and upper thresholds will be reviewed by a team of expert genealogists to determine which are true links. By manually reviewing all pairs that fall between the upper and lower thresholds, we expect the recall and precision of the labeled datasets to be high. As we identify additional links over time, we will add those links to the labeled datasets. Once the labeled datasets have been created, we will use them in the development and evaluation of more advanced algorithms. We are considering making available our labeled sample PRF dataset as a benefit to others developing and evaluating record linkage algorithms. We expect that this dataset would contain approximately 600 pedigrees chosen at random totaling nearly one million individuals, with the links labeled. Depending upon the level of interest, other datasets could follow. Anyone interested in the possibility of obtaining labeled datasets for research purposes should contact Dallan Quass.



Searching or blocking is used to reduce the number of comparisons of record pairs by bringing potentially linkable record pairs together. A good attribute variable for blocking should contain a large number of attribute values that are fairly uniformly distributed and such an attribute must have a low probability of reporting error. Errors in the attributes used for blocking can result in failure to bring linkable record pairs together. For text attributes, various phonetic codes have been derived to avoid effects of spelling and aural errors in recording names. Common phonetic codes include Russell-Soundex and NYSIIS. These codes were optimised for specific populations of names and a specific type of English

pronunciation. Some commercial systems provide tools to derive phonetic codes for specific populations worldwide.



Conclusion and Future work :

Adaptive learning where the comparator function is learnt has also recently been proposed. Predictive models from machine learning such as bagging methods and SVMs have been suggested for learning the match/non-match decision function. Other learning methods for learning the comparator functions have also been proposed. A direct comparison with the Fellegi-Sunter approach has not yet been done but would be worthwhile. Another area of interest is avoiding the need to sort large datasets for blocking. This can be done by using recent developments in high dimensional similarity joins. These techniques use clever data structures
to store records so that good candidates for matching are stored together based on the agreed distance or probabilistic measure.

REFERENCES:

[1] W. Alvey and B. Jamerson, editors. Record Linkage Techniques – 2010. Federal Committee on Statistical Methodology, Washington, D.C., 2010.
[2] M.G. Arellano, G.R. Peterson, D.B. Petitti, and R.E. Smith. The California Mortality Linkage System (CAMLIS). Am. J. Pub. Health, 74:1324–30, 2010.
[3] Inc. Arkidata. Arkistra, 2003.
[4] A.L. Avins, W.J. Woods, B. Lo, and S.B. Hulley. A novel use of the link-file system for longitudinal studies of HIV infections: a practical solution to an ethical dilemma. AIDS, 7:109–113, 1993.
[5] R. Baldoni, C. Cappiello, C. Francalanci, B. Pernici, P. Plebani, M. Scannapieco, S.T. Piergiovanni, and A. Vigirillito. Dl4.a: Design and definition of the cooperative architecture supporting data quality, December 2001.
[6] F.H. Barron and B.E. Barrett. Decision Quality Using Ranked Attribute Weights. Management Science, 42(11):1515–1523, 1996.
[7] T.R. Belin and D.B. Rubin. A Method for calibrating false-match rates in record linkage. Journal of the American Statistical Assocation, 90(430):694–707, June
1995.
[8] G.B. Bell and A. Sethi. Matching Records in a National Medical Patient Index. Communications of the ACM, 44(9):83–88, 2001.
[9] P. Bertolazzi, L. DeSantis, and M. Scannapieco. Automatic Record Matching
in Cooperative Information Systems. In Int. Workshop on Data Quality in Cooperative Information Systems, Jan 2003.
[10] M. Bilenko and R.J. Mooney. Learning to Combine Trained Distance Metrics for Duplicates Detection in Databases. Technical Report AI-02-296, University of Texas at Austin, Feb 2002.
[11] Vinayak R. Borkar, Kaustubh Deshmukh, and Sunita Sarawagi. Automatic
segmentation of text into structured records. In SIGMOD Conference, 2001.
[12] F. Borst, F-A. Allaert, and C. Quantin. The Swiss Solution for Anonymous Chaining Patient Files. MEDINFO 2001, pages 1239–1241, 2001.

[13] Andrew Borthwick. Choicemaker Technologies, Inc., 2002.
[14] K.J. Brameld, C.D.J. Holman, M. Thomas, and A.J. Bass. Use of a state data bank to measure incidence and prevalence of a chronic disease: end-stage renal failure. American Journal of Kidney Disease, 34(6):1033–1039, 1999.
[15] K.J. Brameld, M. Thomas, C.D.J. Holman, A.J. Bass, and I.L. Rouse. Validation of linked administrative data on end-stage renal failure: application of record linkage to a 'clinical base population'. Aust. NZ J. of Public Health, 23:464–467.

About The Authors

(1) P CHANDRASHEKAR REDDY. Completed M.Tech from OU in a Specialization Computer Science and Engineering.. Presently Working as Asst. Proff in MREC, JNTU, and Hyderabad. Area of Specialization are Data Mining, Data Warehouses, DBMS, software engineering..

(2) Y Sushma . Completed B.Tech from JNTU, HYd in a Specialization Electronic and Communication Engineering.. Presently pursuing M.Tech in Computer Science and Engineering in MRCET, JNTU, Hyderabad. Area of Specialization are Data Mining, Data Warehouses, DBMS, software engineering.. and Computer Networks.

(3) B Aruna . Completed Msc from OU , HYd in a Specialization of Computers.. Presently pursuing M.Tech in Computer Science and Engineering in MRCET, JNTU, Hyderabad. Area of Specialization are Data Mining, Data Warehouses, DBMS, software engineering.. and Computer Networks.