Schema Matching using Machine Learning

Literature Review

Schema matching is the problem of identifying which attributes in two distinct schemas refer to the same or simiar data and can be consequently either mapped to one another or used interchangeably. Due to the vastitude of databases and the variations in the way their schemas are written, the same attribute can be denoted by different names, which makes it harder to find a match and subsequently, extract common information across databases. The task has been tackled in various ways and in this review, we present the ones most relevant to our approach towards this subject. The earliest work has been presented first, followed chronologically by others.

Semantic Integration in Heterogenous Databases using Neural Networks [2]
This paper discusses a method of schema matching that employs neural networks in order to learn which subset of attributes of a database A can an attribute of the database B belong to and what their percentage of similarity is. The method is different from others in that it makes use of a learning-based approach rather than a rule-based one. Features of each column are extracted using a database specific parser and these features, represented as vectors with each value lying in the range (0,1) are used as identifiers for that column. Here, the assumption that attributes that represent the same or similar real world data will have similarity in their context and structure has been made. Once the features for each column have been prepared, they are clustered together using a self organizing map. The purpose is to group attributes that represent similar values (such as SSN and Employee ID, both of which have a fixed length, are unique and do not contain zero values, among other similarities) together into a single cluster. Number of desired clusters are taken as input from the user. Once clusters have been prepared, their centers are calculated and then used to train a single hidden layer neural network with M outputs neurons (M = number of clusters). To check which attributes of database B are similar to which attributes of the training database A, attribute feature vectors from B are passed to the neural network. The output gives the similarity percentage of the attribute with each cluster. The features created depend on the attribute's data type.

While this method, known as SemaInt, provides the user with a similarity mapping of each attribute in one schema with a set of attributes in another, it does not take into account the fact one might map to a set of others as well. For example, the attribute "Address" might map to the set "Apt No, St Name, City". It also does not leverage the schema semantics i.e. it does not take into account the name of the attribute when computing its feature vector.

Reconciling Schemas of Disparate Data Sources: A Machine Learning Approach [3]
The authors in the paper Reconciling Schemas of Disparate Data Sources: A Machine Learning Approach have explicated the need of companies to uniformly access multiple sources of data together. In data integration systems, the user interacts with only one integrated data source which internally communicate with other sources to get the information. So, it keeps the users away from the details of the schemas of various remote sources. This internal communication of the sources requires semantic schema mappings between integrated and the other data sources. This paper solves the schema matching problem using LSD (Learning Source Description) approach which is nothing but an extension of the Machine Learning classification problem. It has two phases: Training Phase and Matching Phase. In the Training phase, firstly, the authors manually specify schema mappings (i.e. provide labels to the schema) for some sources and extract some of their data. Then they train the two base learners: Nave Bayes and Name Matcher using the extracted data and the labels assigned manually. After training, cross-validation approach is used and each learner is given a weight. In the matching phase, the match of each remaining data using the trained base learners is found and predictions from both learners are then combined into a single prediction using the weights calculated in the training phase. This whole approach is constrained only for one to one mappings, for eg: Name can be mapped to Employe_name but not to both F_name and L_name. This approach differs from SemaInt in that authors are manually matching a subset of the schemasand using this information to train their systems, as opposed to extracting descriptive features for each attribute.

One of the major drawbacks of this approach is again that the possibility of one attribute mapping to a set of attributes has not been considered. Also, a large dataset is required to perform the necessary training of the classifiers.

Database Schema Matching Using Machine Learning with Feature Selection [1]
This paper is discussing about a tool called Automatch for automating the schema matching process. It matches each attribute of one client to another client schema. It uses feature selection to determine what attributes of the global dictionary will be more relevant to a given schema. This approach consists of a dictionary which is created by using schema examples and tuned by domain experts. Dictionary includes various clusters of attributes say R1, R2, R3 etc. It compares attributes of one schema (S1, S2, S3etc) with each of the dictionary attributes (R1, R2, R3 etc) and assign a weight based on probability formula of symmetry. The same is repeated with another schema and a path from schema 1 to schema 2 via the dictionary is chosen. The Minimum Weight Path determines which attribute of schema 1 is closely aligned with which schema 2 attribute.

While this method improves on its predecessors by including one-to-one attribute matching rather than just matching one attribute with a set of possible attributes, it still has the same problem that it does not consider the possibility of one attribute matching to a set of attributes.

From the approaches shown above, it can be concluded that while each approach has its own strengths, none of them consider the crucial case of an attribute matching to a set of attributes. While there are some methods that do take this into account [4], they have not been considered here as they do not use machine learning. In this project, we propose to overcome this by manually creating a dictionary of a set of possible mappings such as: [Address:Address, St_name, Apt_name, city, Apt Number, Street, Apt_No, Add, Name:Name, First_Name, Last_Name, Fname, Lname] and more and in the similar manner as other papers, prepare a feature vector of each column. These features will include all those extracted in the SemaInt method along with one additional feature which will have values k between 0 and No_of_keys_in_dictionary, where the value 0 will mean that the attribute does not exist in the dictionary and the values 1 to No_of_keys_in_dictionary will denote which key index the attribute belongs to (1-Address, 2-Name etc.). Once this is done, all attributes belonging to group k will be put together in the cluster k. Thus all attributes belonging to the key-value pair for address will be in cluster 1 and so on. Once these clusters have been prepared, each attribute for the test schema will be put into the most relevant cluster and within the cluster, one to one matching will provide the similarity percentage between each attribute.

References

[1] Jacob Berlin and Amihai Motro. Database schema matching using machine learning with feature selection. In *Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, CAiSE '02, pages 452–466, London, UK, UK, 2002. Springer-Verlag.

[2] AnHai Doan, Pedro Domingos, and Alon Y. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, SIGMOD '01, pages 509–520, New York, NY, USA, 2001. ACM.

[3] Wen-Syan Li and Chris Clifton. Semantic integration in heterogeneous databases using neural networks. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 1–12, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[4] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 49–58, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.