Schema Matching using Machine Learning

Tanvi Sahay
tsahay@cs.umass.edu

Ankita Mehta
amehta@cs.umass.edu

Shruti Jadon
sjadon@cs.umass.edu

***Abstract*—Schema Matching is a method of finding attributes that are either similar to each other linguistically or represent the same information. In this project, we take a hybrid approach at solving this problem by making use of both the provided data and the schema name to perform one to one schema matching and introduce creation of a global dictionary to achieve one to many schema matching. We experiment with two methods of one to one matching and compare both based on their F-scores, precision and recall. We also compare our method with the ones previously suggested and the highlight differences between them.**

***Keywords*—Schema Matching, Machine Learning, SOM, Edit Distance, One to Many Matching, One to One Matching**

I. Introduction

The schema of a database is the skeleton that represents its logical view. In other words, a schema describes the data contained in a database, with the name of each attribute in a relation and its data type contained in the relation's schema. Any time the different tables maintained by a peer management system need to be linked to each other or when one branch of a company is closed down and all its data needs to be redistributed to the database maintained by other branches or when one company takes over another company and all data of the child comapny needs to be integrated with that of the parent company, the need to match schemas of multiple relations with each other arises. In basic terms, schema matching can be explained as follows: Given two databases $X(x_1, x_2, x_3)$ and $Y(y_1, y_2, y_3)$ with x_n and y_n representing their attributes resepectively, we match a schema attribute to another either if it is linguistically similar(has a similar name) or if it represents the same data. Consider the Tables I and II. Here, the ideal schema mappings would be: *FName* + *LName* = *Name*, *Major* = *Maj_Stream* and *Address* = *House No* + *St Name*.

TABLE I. Students

FName	LName	SSN	Major	Address
Shruti	Jadon	123-aaa-aaaa	Computer Science	1xx Brit Mnr
Ankita	Mehta	234-bbb-bbbb	Mathematics	2xx Boulders
Tanvi	Sahay	456-ccc-cccc	Political Science	3xx N Pleasant St

TABLE II. Grad-Students

Name	ID	Maj_Stream	House No	St name
Shruti Jadon	123aaa	CompSci	1xx	Brit Mnr
Ankita Mehta	23bbb4	Math and Stats	2xx	Boulders
Tanvi Sahay	45cccc	PoliSci	3xx	N Pleasant St

Over the years, researchers have faced several issues when trying to automate the process of matching schemas of different relations. Because the schemas are created by human developers and are pertinent to a particular domain, human intervention is often required at one or multiple stages of the process to ensure proper schema matching, which makes this taks quite labor intensive. The aim of automated schema matching is to reduce the involvement of a domain expert in the process to a minimum. Majorly, schema matching can be divided into two parts - one to one matching, where one attribute of table 1 matches with only one attribute of table 2 and one to many matching, where one attribute of table 1 may map to a combination of several attributes of table 2. While one to one matching has been successfully automated using sophisticated machine learning techniques as well as by exploiting the schema structure, performing one to many schema matching still requires some form of human intervention. In general, matching can be done by taking into account either the data contained in the relations or the name of the attributes or both.

In this project, we explore two methods of performing one to one matching and suggest a new method of one to many mapping which is different from the ones that have been employed before. For one to one matching, we consider two appoaches, both based on utilizing a set of features to find similar attributes. In the first method, called centrid method, we cluster similar values of one table together into groups and compare each attribute of the second table with each cluster, to find the one that best matches with it. In the second method, called the combined method, we cluster attributes of both tables together to form groups containing fields from both tables. The centroid method, as we will see in the future sections, ensures that every attribute in the second table matches with at least one attribute in the first table. The combined approach on the other hand still has the possibility of an attribute in one table not matching with any other attribute in the second table. Each method will be discussed in more detail in the future sections and their tradeoffs as well as their performance with existing techniques will be compared as well. In addition to these techniques, we will discuss a new way of taking care of one to many matchings with minimum requirement of an external expert.

II. Previous Work

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. This approach consists of a global 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.

Semantic Integration in Heterogenous Databases using Neural Networks*[2]*

This paper implemented schema matching using Machine Learning approach. It extracts the features of each column by using only their data values and these features, represented as vectors with each value lying in the range (0,1) are used as identifiers for that column. Then they are clustered together using a self-organizing map and their cluster centres are calculated. Using these cluster centres single hidden layer neural network with M outputs neurons (M = number of clusters) is trained and then tested with output as the similarity percentage of the attribute with each cluster.

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.

Corpus-based Schema Matching*[3]*

In this paper, schema matching is applied on the corpus containing multiple schemas that model similar concepts. It trains a classifier using the information retrieved from the schema structure, data instances and their contexts. Then using the learned information it predicts the one-to-one mappings.

It doesnt consider the one-to-many and complex mappings and can only be applied on the smaller datasets.

Generic Schema Matching with Cupid*[4]*

This paper explores the internal structure of the schema by combining various techniques viz Linguistic Matching, Structure-based matching constraint-based matching, and context-based matching. It applies TreeMatch algorithm to perform the schema matching by giving tree structure to the schema.

However, it considers both one-to-one and one-to-many mapping but it does not use the information provided by data instances and only explores the schema structure.

iMAP:Discovering Complex Semantic Matches between Database Schemas*[5]*

In this paper, authors have developed various searchers for different data types viz: text, number, date. For example: address = concat(city,state). It exploits the domain knowledge to improve the accuracy of the making searchers.

III. Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Appendix A
Proof of the First Zonklar Equation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Acknowledgment

The authors would like to thank...

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] 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.

[3] Jayant Madhavan, Philip A. Bernstein, AnHai Doan, and Alon Halevy. Corpus-based schema matching. In *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, pages 57–68, Washington, DC, USA, 2005. IEEE Computer Society.

[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.

[5] Qian Ying, Yue Liwen, and Liu Zhenglin. Discovering complex matches between database schemas. In *2008 27th Chinese Control Conference*, pages 663–667, July 2008.