Project Proposal

Schema Merging using Machine Learning

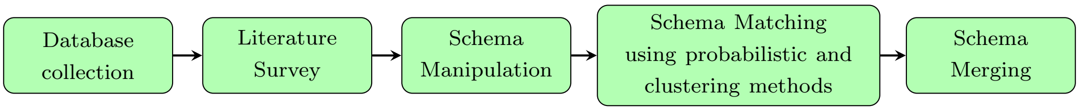
Submitted by:
Ankita Mehta Shruti Jadon Tanvi Sahay

1 Problem Statement

Schema merging refers to the problem of merging two schemas together by matching the attributes which are semantically related and portray similar properties to each other. This problem occurs most frequently when a company takes over another or multiple companies merge to form one and their databases need to be joined together. Often, two attributes can be worded differently but still mean the same thing and represent the same data. When merging them together, the insight that such schema features mean the same is very valuable. For example: one database may have an attribute PID while the other may have ProductID. The two refer to same values but since they are worded differently, pure syntactic matching will not group them together. Also, a database may split the same information into multiple categories while another may keep them clubbed into a single one. For example: (ADDRESS, CITY) and (APT NO, ST NAME, CITY) refer to the same information.

2 Why existing approaches fail

The most common approach adopted by companies today is employing database experts to manually match schemas of multiple databases and convolve them to a single common schema. This works for small databases but as the data dimension increases, this process becomes time consuming and expensive. Some methods have tried to tackle this problem by using a rule-based approach (ARTEMIS) while others have employed machine learning techniques such as SemiInt and Automatch. While they address the problem of semantic matching and provide a one-to-one match, they do not take into consideration the possibility of many-to-one schema mapping.



3 Proposed solution

We will attempt to solve the many-to-one problem using a rule-based approach. A dictionary of mappings will be created and each database schema will be checked for such mappings. In case a valid combination is possible, the data columns will be combined together and stored separately, to be used for comparison with the second client database. This schema manipulation will be performed only for the child client and not the parent client. For example: one possible mapping can be: (APT NO + ST NAME) $--->$ ADDRESS. Then, the child client database will be scanned for the presence of either of the two instances mentioned above. If instance 1 i.e. (APT NO, ST NAME) exists, it will be mapped to ADDRESS and stored separately. This way, if the parent client has ADDRESS as one attribute instead of 2, proper mapping will still be performed. This is just an initial approach and has the drawback that it cannot do a one-to-many mapping without substantially increasing the disk usage.

Once schema manipulation for each client has been done, schema matching will be performed. Two approaches will be considered for this. In the first approach, one-to-one matching using probabilistic methods will be explored. A matching score will be provided to each pair and pairs with the highest probabilities will be selected as "good matches". In the second approach, clustering will be performed. Since the database is considered to have a high number of attributes, clustering should also act as a primary form of dimensionality reduction. Cluster centers of each cluster for both client databases will then be matched with each other using the same probabilistic models introduced in the first method. Once clusters have been finalized, attribute-to-attribute matching will be similarly performed to obtain the "good matches".

4 Dataset to be used

The datasets will be chosen on the premise that one of them will be the parent database and the rest will be the child databases. All data will be chosen from a single domain, such as medicine, tourism, shopping stores etc.