**DSE 203 10.28.2017**

**Schema Matching**

Name-based matching -> based on a synonym dictionary & string similarity

Instance-based matching -> recognizers (dictionaries, regex, simple rules), overlap map (overlap among attributes), classifiers (using learning techniques)

Recognizers ie. country names, person names (w/1st & last name dictionaries)

Overlap map -> typically applied to attributes whose values are drawn from some finite domain, Jaccard measure is commonly used

Classifiers -> builds classifiers one 1 schema & uses them to classify the elements on the other schema ie. Naïve Bayes, decision tree, rule learning, SVM

* Use +/- examples for training, computer similarity scores, and determine confidence in matching
* Designer decides which schema should play the role of schema S -> typically chooses the mediated schema to be S, so that can reuse the classifiers to match the schemas of new data sources (but can do it both ways)

Combining Match Predictions

Average combiner -> when we do not have any reason to trust matcher over the others

Maximum combiner -> when we trust a strong signal from matches ie. if a matcher outputs a high value, we are relatively confident that the 2 elements match

Minimum combiner -> when we want to be more conservative

More complex types of combiners -> weighted-sum combiners (ie. lin reg, log reg., can train to determine weights of each matcher), the combiner itself can be a learner (ie. DT, log reg.)

Schema.org determines standards for web publishing, get basic schema of common things

In Schema matching -> is this column or combo of columns that column

**Schema Mapping**

Want to create an integrated view

How do we know which elements of the mediated (target) schema gets it data supply from which elements of the source schema?

Not always 1 equation that works for all, would need to identify that there are 2 subgroups ie. Professor & student, take least cost join for students, create unions

Example:

select P.HrRate \* W.hrs

from **PayRate P**, **WorksOn W**

where P.Rank = W.ProjRank

UNION ALL

select Sal

from **Professor**

Have the translation of every value into target value ie. nulls

**Datalog**

Relations = predicates

Tuples = atoms

Ie. Purchase(“joe”, “bob”, “Nike Town”, “Nike Air”)

Head (output) :- (if) atom, atom1, …

Ie. construct a result set name it PerformingComp & query from Company based on an arithmetic comparison

PerformingComp(name) :- Company(name, sp, c), sp > $50

* All vars in the head must appear in the body

If each of the atoms in the body is in the db, then the tuple for the head is in the result

Q5() -> a union in datalog

* Unions remove duplicates

Unsafe query -> might return the entire database

Q(X,Y,Z) :- R1(X,Y) & X < Z not safe (b/c no ideas what Z is)

Q(X,Y,Z) :- R1(X,Y) & NOT R2(X,Y, Z) not safe (b/c can’t even process, every var that appears anywhere in the query must appear also is a relational, non-negated atom in query)

memberOf(X,S) -> returns T if term X is a member of set S

concatenation (|) -> adds a member to a set ie. sum(X | S, A) :- sum(S1, A1), A = A1 + X (recursive fxn)

**More Schema Mapping**

Rewrite a user query expressed in the mediated schema into a query expressed in the source schema

Find a query Q’ that uses only the source relations, such that Q’ is included in Q & Q’ provides all possible answers to Q given the sources

Global as View (GAV): query -> rewriting by view unfolding (rule unfolding) -> remove redundant subgoal in body -> query containment

* What does this imply? -> do not have to revisit different dbs, but can revisit the cached results
* Will not have to go to db1 and db3 but will have to go to db2
* Need to know is the query a subset of another query
* Downfall: every time data updates you update your views

Local as view (LAV): query is placed vs the mediator’s schema & needs to be transformed by “inverting” the mapping to construct queries vs. the sources