Semantic Schema Matching Without Shared Instances

Jeffrey Partyka, Latifur Khan, Bhavani Thuraisingham
Department of Computer Science, The University of Texas at Dallas
800 West Campbell Road
Richardson, TX, 75083-0688, United States
{jlp072000, lkhan, Bhavani.thuraisingham
} @utdallas.edu

Abstract

Semantic heterogeneity across data sources remains a widespread and relevant problem requiring innovative solutions. Our approach towards resolving semantic disparities among distinct data sources aligns their constituent tables by first choosing attributes for comparison. We then examine their instances and calculate a similarity value between them known as entropy-based distribution (EBD). One method of calculating EBD applies a state-of-the-art instance matching strategy based on N-grams in the data. However, this method often fails because it relies on shared instance data to determine similarity. This results in an overestimation of semantic similarity between unrelated attributes and an underestimation of semantic similarity between related attributes. Our method resolves this using clustering and a measure known as Normalized Google Distance. The EBD is then calculated among all clusters by treating each as a type. We show the effectiveness of our approach over the traditional N-gram approach across multi-jurisdictional datasets by generating impressive results.

1. Introduction

The problem of information integration has experienced a number of manifestations since its inception, which resulted from the meteoric popularity of relational databases after the 1960's. However, the core of this problem has always been the need to consolidate heterogeneous data sources under a single, unified schema. Over the last few decades, a tremendous amount of effort has been expanded to discover novel information integration strategies.

In this paper we attempt to compare two pairs of data sources by examining the instances of compared tables; the first pair of data sources contains tables describing similar models of transportation network over multiple jurisdictions, while the other pair contains tables detailing varying geographic features. The data sources contain large variations in the geographic areas covered, the number of attributes, and the number of instances.

To measure instance similarity between compared attributes we will attempt to match the respective distributions of their representative types. A type will be defined as a common representation of a group of related pieces of data. Once all types for the compared attributes have been accounted for, the semantic similarity between the attributes is calculated using a measure known as entropy-based distribution (EBD). EBD is based on the ratio of the conditional entropy within the types extracted for a pair of compared attributes with the entropy over all types.

We examine two different instance similarity algorithms. The first examines keywords in the compared attributes and extracts subsequences of their characters known as N-grams. The idea behind this method is that keywords that share more N-grams are more semantically similar to one another. However, this idea often proves to be incorrect in situations where few shared instances exist over multiple jurisdictions. The second approach, which we will dub as the TSim algorithm, executes instance matching by applying a similarity metric known as the Normalized Google Distance (NGD). The end result is a group of distinct clusters (hence types), each of which contains a unique set of keywords related to each other through common semantic features. The similarity between the attributes is then computed by calculating the EBD. Because we do not have to depend on shared N-grams for semantic similarity, our instance matching

978-0-7695-3800-6/09 $26.00 © 2009 IEEE
DOI 10.1109/ICSC.2009.64

algorithm can derive a more realistic measure of the implicit semantics existing between any given pair of attributes from distinct data sources.

Our main contributions are as follows. First, we display the inadequacies of the N-gram approach by testing it on multiple datasets and highlighting its inability to identify correct semantic correspondences between attributes due to its reliance on shared instances. Second, we propose a new algorithm, called TSim, that derives semantic similarity between attributes of compared tables without the need for shared instances. This is accomplished through K-medoid clustering of the instance data associated with the attributes into distinct semantic types, with the help of NGD. Finally, we show the effectiveness of our approach relative to the traditional N-gram method through lucid results on two separate datasets.

The rest of this paper is organized as follows. In section 2, we discuss an overview of related work. Section 3 states the problem to be solved and our proposed solution. Section 4 presents in detail the TSim algorithm alongside the current, state-of-the-art approach that depends on shared N-grams. In Section 5 we present results. Finally, in section 6, we outline our future work.

2. Related work

A number of schema matching publications [1,2,3,4,5] tailored to the database community and instance-based ontology matching [9,10] from the ontology matching community, influenced our work. The survey of approaches to automated schema matching by Rahm and Bernstein[1] includes a taxonomy which uses several criteria to categorize the matching approaches such as schema and instance based methods, element-level and structure-level methods, and linguistic and constraint-based methods. Dai, Koudas et al. [2] discuss instance-based schema matching using distributions of N-grams among compared attributes. Bohannon et. al[3] investigate contextual schema matching, in which selection conditions and a framework of matching techniques are used to create higher quality mapping between attributes of compared schemas. Warren and Tompa [4] propose an iterative algorithm that deduces the correct sequence of concatenations of column substrings in order to translate from one database to another without the use of a set of training instances.

Our paper presents an innovative instance matching algorithm that possesses a number of advantages over the N-gram approach proposed by Dai, Koudas et al. First, our new instance matching approach leverages clustering of types for use on distinct keywords found between compared attributes. This approach is better able to capture the semantics of comparisons between attributes because words contain more implicit semantic information than N-grams. Using words, we can reference external data sources that allow for distance metrics to determine word relatedness. In general, this cannot be done with N-grams because they are usually just parts of words. Second, our new instance matching algorithm is flexible enough to allow for different types of semantic distance measures to be used. Treating the semantic distance measure as a pluggable component allows for a wider variety of experiments to be performed on a given instance set, which in turn leads to a better understanding of the kinds of semantic distance measures that best suits a particular type of data. Finally, the use of N-grams for instance similarity between data sources sometimes generates misleading results, especially in cases where data of different languages but similar semantics is being compared.

Since we use Google distance to calculate similarity there is some relevant work. Gligorov et al. [7] apply Google distance [6] to clearly distinguish between pairs of words which are not semantically related and pairs of words that possess a close semantic relation. However, our approach differs from their approach in the following ways. First, Gligorov et al. use Google distance to automatically assign appropriate weights (or importance) to the similarity between concepts associated with a concept hierarchy for the purposes of ontology matching. On the other hand, we use Google distance as a measure to aid in the construction of cohesive clusters containing similar-themed keywords which are then used to perform automated schema matching between individual concepts. Next, Gligorov et al. do not consider instance-based matching; they purely exploits concept labels while our idea of matching is based on the instances associated with the compared concepts.

3. Problem statement and proposal

3.1 Definitions

First, we will provide definitions that will assist in defining the problem and describing TSim.

Definition 1 (attribute) *An attribute of a table T, denoted as att(T), is defined as a property of T that further describes it.*

Definition 2 (instance) *An instance x of an attribute att(T) is defined as a data value associated with att(T).*

Definition 3 (type) *A type t associated with attribute att(T) is defined as a class of related entities grouped together.*

In figure 1 below, the two attributes for the given table are roadName and City, and two instances from the roadName attribute are “Johnson Rd.” and “School Dr.”.

roadName	City
Johnson Rd.	Plano
School Dr.	Richardson
Zeppelin St.	Lakehurst
Alma Dr.	Richardson
Preston Rd.	Addison
Dallas Pkwy	Dallas

Figure 1. Sample table containing two attributes and six instances

3.2. Problem statement

Given two data sources, S_1 and S_2, each of which is composed of a set of tables/relations where $\{T_{11}, T_{12}, T_{13}\ldots T_{1M}\}$ S_1 and $\{T_{21}, T_{22}, T_{23}\ldots T_{2N}\}$ S_2, the goal is to determine the semantic similarity between S_1 and S_2. This is done by comparing the respective attribute names and attribute values, or instances, between the tables from S_1 and those from S_2. S_1 and S_2 may be derived from any domain. Additionally, S_1 and S_2 may vary in regards to the number of constituent tables, the number of attributes and instances within a given table.

3.3. Proposed solution

We present two separate instance matching algorithms that generate semantic similarity values between compared attributes in different tables. The first, based on the ideas of mutual information and entropy, extracts features consisting of sequences of characters with length N known as N-grams from the values of the compared attributes [2]. Each N-gram extracted is considered a distinct value type, and the ratios of value types originating from each attribute is determined to be their overall semantic correspondence. While this method can be successful for certain datasets, it can produce incorrect results for others, such as a multi-jurisdiction dataset, where no/few shared instances exist. Section 4 outlines in detail one such situation. The second instance matching algorithm, based on the extraction and clustering of semantically relevant keywords as types, treats distinct keywords extracted from compared attributes, rather than N-grams, as features. Further details describing the algorithm are described in Section 4.3. However, it is our intention to clearly show that the use of TSim on distinct keywords is better able to capture the true semantics that exist between compared attributes contained within tables..

It is assumed that we perform 1:1 comparisons between attributes from distinct tables and data sources. After calculating a semantic similarity value between compared attributes, we will repeat the process for all compared attributes between the tables. Next, a final similarity value between the tables is calculated.

4. Matching algorithm: semantic similarity between two tables

4.1. Instance similarity using N-grams

Instance matching between two concepts involves measuring the similarity between the instance values across all pairs of compared attributes. This is accomplished by extracting instance values from the compared attributes, subsequently extracting a characteristic set of N-grams from these instances, and finally comparing the respective N-grams for each attribute. N may be any number, so during all of our experiments involving N-grams in this paper, the value of N was set equal to 2.

4.1.1. Feature Extraction of N-grams

We extract distinct N-grams from the instances and consider each unique N-gram extracted as a type. A type in this context is defined as 2-gram represented by an identifying string of length 2. As an example, for the string "Locust Grove Dr." that might appear under an attribute named Street for a given concept, some 2-grams that would be extracted are 'Lo', 'oc', 'cu', 'st', 't ', 'ov', 'Dr' and so on. Since each of these 2-grams are different, each one would represent a distinct type.

4.1.2. Measuring attribute similarity

N-gram similarity is based on a comparison between the concepts of entropy and conditional entropy known as Entropy Based Distribution (EBD):

$$EBD = \frac{H(C \mid T)}{H(C)} \qquad (1)$$

In this equation, C and T are random variables where C indicates the union of the attribute types C_1 and C_2 involved in the comparison (C indicates "column", which we will use synonymously with the term “attribute”) and T indicates the type, which in this case is a distinct N-gram. EBD is a normalized value with a range from 0 to 1.

Entropy is defined as the measure of the uncertainty associated with a random variable, whereas conditional entropy is defined as the uncertainty associated with one random variable given the value of a second random variable. Conditional entropy is defined as follows:

$$-\sum_{t \in T} \sum_{c \in C} p(c,t) \log\ p(c \mid t) \qquad (2)$$

Our experiments involve 1:1 comparisons between attributes of compared tables, so the value of C would simply be C_1 U C_2. H(C) represents the entropy of a group of types for a particular column (or attribute) while H(C | T) indicates the conditional entropy of a group of types. For more details regarding the usage of EBD and its mathematical derivation, please see our previous work[8].

4.2. Motivation For TSim

4.2.1. Problems With N-grams as a Measure For Semantic Similarity

N-grams are susceptible to generating misleading results. For example, if an attribute named 'City' associated with a table from S_1 is compared against an attribute named 'ctyName' associated with a table from S_2, the attribute values for both concepts might consist of city names from different parts of the world. 'City' might contain the names of North American cities, all of which use English and other Western languages as their basis language, while 'ctyName', might describe East Asian cities, all of which use languages that are fundamentally different from English or any Western language. Using human intuition, it is obvious that the comparison occurs between two semantically similar attributes. However, because of the tendency for languages to emphasize certain sounds and letters over others, the extracted sets of 2-grams from each attribute would very likely be quite different from one another. For example, some values of 'City' might be "Dallas", "Houston" and "Halifax", while values of 'ctyName' might be "Shanghai", "Beijing" and "Tokyo". Based on these values alone, there is virtually no overlap of N-grams. Because most of the 2-grams belong specifically to one attribute or the other, the calculated EBD value would be low. This would most likely be a problem every time global data needed to be compared for similarity.

4.2.2. Overview of the TSim Algorithm

To overcome the problems of the N-gram approach, we need a method that is free from the syntactic requirements of N-grams and uses the keywords in the data in order to extract relevant semantic differences between compared attributes. This method, known as TSim, extracts distinct keywords from the compared attributes and determines their types by leveraging K-medoid clustering to group together keywords of the same type based on a semantic distance metric known as the Normalized Google Distance (NGD). The EBD is then calculated by comparing all instances of keywords representing each type, where a cluster is considered a distinct type.

4.3. The TSim algorithm

We determine semantic similarity between two separate data sources through K-medoid clustering of the keywords extracted from the compared attributes. The distance metric used in assigning keywords to clusters is known as Normalized Google Distance.

4.3.1. Normalized Google Distance

Before describing the process in detail, NGD must first be formally defined:

$$\mathrm{NGD}(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log M - \min\{\log f(x), \log f(y)\}} \qquad (3)$$

In this formula, f(x) is the number of Google hits for search term **x**, f(y) is the number of Google hits for search term **y,** f(x,y) is the number of Google hits for the tuple of search terms **xy**, and M is the number of web pages indexed by Google. For more information about NGD, consult the work by Gligorov et al[7].

4.3.2. Clustering the Keywords

Once the keyword list for a given attribute comparison has been created, all related keywords are grouped into distinct clusters. From here, we calculate the conditional entropy of each cluster by using the number of occurrences of each keyword in the cluster, which is subsequently used in the final EBD calculation between the two attributes. The clustering algorithm used is the K-Medoid algorithm, which is described in the next section.

4.3.3. The K-Medoid Algorithm

The K-medoid algorithm begins by first determining the number of clusters, dubbed K. This is based on the size of $L_{keywords}$ for each attribute comparison. Second, exactly one keyword from the list is assigned to each of the K clusters in a process called initial seeding. The keywords assigned to the clusters in this step are known as medoids. Third, we assign each keyword in $L_{keywords}$ that is not a medoid to the cluster to which it is most semantically related, while subsequently determining if any cluster medoids need to be recomputed. To do this, we need to use the pairwise NGD values list between the keyword to be assigned to a cluster and all keywords already assigned to that

same cluster. Finally, after all keywords have been assigned to clusters, we determine if the medoid for any cluster needs to be recomputed. This is accomplished by examining each of the keywords in a particular cluster and computing an NGD summation between a single keyword in that cluster and all other words in that cluster. The keyword in that cluster that produces the lowest NGD summation will be assigned as the new medoid for that cluster. If no medoids have changed in any cluster, then the K-medoid algorithm is finished, and control proceeds to the calculation of the EBD between the compared attributes. However, if at least one medoid has changed in a particular cluster, then we begin a new clustering iteration.

5. Experiments

We now present the experiment that we conducted regarding matching between distinct data sources in the GIS domain.

5.1. Experimental Setup

Two separate datasets from the GIS domain were used to evaluate the performance of TSim. The first dataset was created from instance data of the Road and Ferries package of a GIS data model known as GDF (Geographic Data Files). The second dataset details a wider assortment of GIS location features across the United States and their associated data beyond merely transportation networks. Some of the location features in this dataset include flight schools, piers, navigable waterways and Indian lands. For both sets of data, the number of attributes and instances vary widely; for example, in the GIS location dataset, the Flight Schools table has the fewest number of attributes (27) and the Piers table has the most (76). Because data from several different areas of the United States were employed in our experiments, we effectively created a *disjoint, multi-jurisdictional environment.* Table 1 below displays a summary of the relevant information regarding the data involved in our experiments with both datasets.

Table 1. Description of (a) transportation dataset & (b) GIS Location Dataset

Table Name	No. of Attributes	Area(s) Modeled	No. of Instances
Road(S_1), Road(S_2)	18,11	Fort Collins, CO Dallas, TX	9851, 5224
Ferry(S_1), Ferry(S_2)	3,8	Seattle, WA	24,42
Traffic Area (S_1), Traffic Area (S_2)	24,26	Virginia	329, 108
Residential Area (S_1), Address Area (S_2)	15,10	New Jersey, Texas	4263, 2122

Table Name	No. of Attributes	No. of Instances
Flight Schools (S_1), Flight Schools (S_2)	27	4653 and 4653, respectively
Schools (S_1), Schools (S_2)	41	57728 and 56730, respectively
Piers (S_1)	76	6159
Indian Lands (S_1)	52	15852
Ports (S_2)	56	4534
NavWaterways (S_2)	30	6879

Table 2a and 2b. Comparison of EBD values generated by the N-gram method and TSim for correct attribute correspondences. In table 2a (left), the N-gram method underestimates the similarity, and in table 2b (right), N-grams overestimate the similarity

	EBD from TSim	EBD from N-Gram
Flight Schools.BUSINESSNM – Schools.NAME	.691	.117
NavWaterways.Name – Piers.Name	.633	.083
NavWaterways. WTWY– Piers.WTRWY	.981	.095
Ports.COUNTY– Piers.COUNTY	.682	.043
Ports.Port – Piers.port	.735	.437

	EBD from TSim	EBD from N-Gram
Traffic Area.County– Road.City	.145	.525
Traffic Area.County – Ferry.DSP	.298	.560
Road.ADD1– Enclosed Traffic Area.STATE_TOTAL_VMT	.187	.466
Residential Area.countyname – Enclosed Traffic Area.District1	.187	.550
Residential Area.State – Enclosed Traffic Area.STATUS	.343	.808

5.2. Results

An illustration of the tendency of the N-gram method to underestimate the value of correct attribute correspondences relative to TSim and overestimate the value of incorrect correspondences is displayed in table 2a(left) above for the GIS location dataset and in table 2b(right) for transportation dataset. For table 2a, in all five comparisons, the attributes are clearly related (ie: Ports.COUNTY and Piers.COUNTY). However, the N-gram method generates low EBD values for these comparisons (right column of table), while TSim generates high EBD values (left column of table). The reason for this is the inability of the N-gram method to relate two attributes together without the use of shared instances. As long as the compared attribute values are made of widely varying N-gram types, this method will always produce a low EBD value. On the other hand, because TSim does not rely on shared instances to determine semantic similarity, it is able to correctly assign a high EBD score between the attributes. On average, for the five comparisons above, the N-gram method underestimates the EBD score by 77%. Table 3b illustrates that the use of shared instances by the N-gram method can also lead to the exaggeration of similarity scores between unrelated attributes. For example, Traffic Area.County and Ferry.DSP both contain county data including the word "county", but DSP (which stands for 'Destination Port') also contains the names of towns and other geographic features. The N-gram method will match any instances containing the word "county" as well as other instances sharing common words, thus incorrectly raising its EBD computation. On average,

for the five comparisons in table 3b, the N-gram method overestimates the EBD score by 266 %.

The results of the alignment of S_1 and S_2 of the compared tables for both the transportation dataset and the GIS location dataset using TSim are shown in tables 3a and 3b, respectively. Each cell contains the EBD value produced using TSim between a table from S_1 (names listed along the vertical axis of the table) and a table from S_2 (names listed along the horizontal axis of the table).

Table 3a and 3b. EBD values generated between tables of S_1 and S_2 of (a: transportation dataset (left table) (b: GIS location dataset (right table)

	Road	Address Area	Enclosed Traffic Area	Ferry
Road	.553	.225	.276	.503
Residential Area	.210	.552	.433	.407
Traffic Area	.136	.219	.958	.235
Ferry	.127	.237	.424	.564

	Flight Schools	Schools	Ports	NavWaterways
Flight Schools	.720	.615	.532	.503
Schools	.388	.768	.395	.540
Indian Lands	.489	.513	.486	.533
Piers	.496	.489	.633	.616

In table 3a, the EBD values obtained using TSim for the comparisons between Road-Road, Residential Area-Address Area, Traffic Area-Enclosed Traffic Area, and Ferry-Ferry are 0.553, 0.552, 0.958, and 0.564 respectively. Each of these represented the correct correspondences, and TSim identified them as those with the highest semantic similarity. In addition, tables that are semantically dissimilar, such as Ferry-Road and Traffic Area-Address Area were correctly recognized as such by TSim, as scores of .127 and .219 were generated. Similar results are also obtained in table 3b. Both of these datasets illustrate the tendency for the N-gram approach to overestimate incorrect correspondences and underestimate correct correspondences. For example, in table 3a, some of the EBD values produced via TSim for Road-Address Area, Road-Enclosed Traffic Area, and Road-Ferry are 0.22, 0.27 and 0.28 respectively. On the other hand, using the N-gram method, the scores generated for these comparisons were 0.44, 0.43 and 0.48 respectively. The scores were overestimated by 100%, 59% and 71% respectively. In table 3b, using TSim, the EBD values produced for Flight Schools(S_1)-Schools(S_2), Piers-Ports and Piers-NavWaterways are .615, .633 and .616. Using the N-gram approach, the scores generated are .182, .388 and .137. In this case, the N-gram method underestimated the scores by 70.5%, 38.8% and 77.8%, respectively.

6. Conclusion & Future Work

We outlined two algorithms that align distinct data sources using instance similarity. The first algorithm aligns instances between compared attributes by extracting distinct N-grams from them and measuring their semantic similarity by calculating an EBD value. The second algorithm, TSim, determines the semantic types of keywords in compared attributes using clustering and an external data source which leverages the Normalized Google Distance. Future efforts will focus on exploring the possibility of a hybrid instance matching technique that combines selected elements of the N-gram approach and TSim.

7. References

[1] E.Ralun and P. A. Bernstein, "A survey of approaches to automatic schema matching", *VLDB Journal*, vol. V10, pp. 334-350, 2001.

[2] Bing Tian Dai, Nick Koudas, Divesh Srivastava, Anthony K. H. Tung, and Suresh Venkatasubramanian, "Validating Multi-column Schema Matchings by Type," *24th International Conference on Data Engineering (ICDE),* 2008.

[3] P. Bohannon, E. Elnahrawy, W. Fan, and M. Flaster, "Putting context into schema matching." in VLDB, 2006, pp. 307–318.

[4] R. H. Warren and F. W. Tompa, "Multi-column substring matching for database schema translation." in *Proc. VLDB*, 2006, pp. 331–342.

[5] W.S. Li and C. Clifon, "Semint: a tool for identifying attribute correspondence in heterogeneous databases using neural networks," *Data Knowl. Eng.*, vol. 33, no. 1,pp.49-84, 2000.

[6] Rudi Cilibrasi, Paul M. B. Vitányi: The Google Similarity Distance CoRR abs/cs/0412098:(2004)

[7] R. Gligorov, W. Kate, Z. Aleksovski, F. Harmelen: Using Google distance to weight approximate ontology matches. WWW 2007:767-776

[8] Jeffrey Partyka, Neda Alipanah Latifur Khan, Bhavani Thuraisingham and Shashi Shekhar, "Content-based Ontology Matching for GIS Datasets", University of Texas at Dallas (UTD Technical Report # UTDCS-22-08).

[9] Shenghui Wang, Gwenn Englebienne and Stefan Schlobach, "Learning Concept Mappings from Instance Similarity", *Proceedings of the 7th International Semantic Web Conference, ISWC 2008*, LNCS 5318, pp 339-355, 2008.

[10] Christian Wartena and Rogier Brussee, "Instance-Based Mapping Between Thesauri and Folksonomies", In: *Proc. of the 7th International Semantic Web Conference*, ISWC 2008, LNCS 5318, pp 356-370, 2008.