An Evidential Approach to Query Interface Matching on the Deep Web

Jun Hong
School of Electronics, Electrical Engineering and Computer Science
Queen's University Belfast
Belfast BT7 1NN, UK
j.hong@qub.ac.uk

Zhongtian He
School of Electronics, Electrical Engineering and Computer Science
Queen's University Belfast
Belfast BT7 1NN, UK
zhe01@qub.ac.uk

David Bell
School of Electronics, Electrical Engineering and Computer Science
Queen's University Belfast
Belfast BT7 1NN, UK
da.bell@qub.ac.uk

ABSTRACT

Matching query interfaces is a critical step in data integration across multiple Web databases. The problem is closely related to schema matching that typically exploits different features of schemas. Relying on a particular feature of schemas is not sufficient. We propose an evidential approach to combining multiple matchers using Dempster-Shafer theory of evidence. First, our approach views the match results of an individual matcher as a source of evidence that provides a level of confidence on the validity of each candidate attribute correspondence. Second, it combines multiple sources of evidence to calculate the overall level of confidence, reflecting the match results of different matchers. Third, it selects the top k attribute correspondences of each source attribute from the target schema. Finally it uses some heuristics to resolve any conflicts between the attribute correspondences of different source attributes. Our experimental results show that our approach is highly accurate and effective.

1. INTRODUCTION

Web databases are now pervasive, which can be accessed via their query interfaces (usually HTML query forms) only. Query forms provide a natural way for the user to make queries to the underlying databases without using a particular query language. On receiving form-based queries, these databases return query results encoded in HTML, which are then displayed to the user.

Many E-commerce sites are supported by Web databases. In a specific domain (e.g. flight booking, book sales), there are many database-driven Web sites that sell similar products and services. It is a daunting task for the user to visit numerous Web sites individually to search for and compare services or products. Web data integration aims to provide single-point access to a multitude of Web databases, where users need to fill in only a uniform query form, and on receiving a user query the system will automatically make connections to different sites, fill in the local query forms on these sites, submit these forms, combine the query results, and return the combined results to the user.

Matching query interfaces is a critical step in Web data integration, which finds attribute correspondences between the uniform query interface and a local query interface. The problem is closely related to schema matching that takes two schemas as input and produces a set of attribute correspondences between them [1]. Schema matching has been extensively studied (e.g. [1, 2, 3, 4, 5, 6, 7, 8]). These approaches exploit different features of schemas, including structural and linguistic features and data types, etc to match attributes between schemas. Schema matching is inherently uncertain due to lack of complete knowledge about schemas. Relying on a single feature of schemas is not sufficient and the match results of individual matchers are often inaccurate and uncertain. Approaches have been proposed to combine multiple matchers taking into account different features of schemas. A common approach is to apply different weight coefficients to the match results of individual matchers reflecting their different levels of importance, and their weighted values are then added together as the combined match results. However, these weight coefficients are often manually set for a particular domain in a trial and error manner.

We propose an evidential approach to combining the match results of multiple matchers using Dempster-Shafer theory of evidence. First, this approach views the match results of an individual matcher as a source of evidence that provides some degree of belief on the validity of each candidate attribute correspondence. Second, it combines degrees of belief from multiple sources of evidence to calculate the overall degree of belief on the validity of each candidate attribute correspondence, reflecting the match results of different matchers. Third, it selects the top k attribute correspondences of each source attribute from the target schema. Finally it uses some heuristics to resolve any conflicts between the attribute correspondences of different source attributes. Our experimental results show that our approach is highly accurate and effective.

2. DEMPSTER-SHAFER (DS) THEORY OF EVIDENCE

Dempster-Shafer theory of evidence, sometimes called ev-

VLDB '08, August 24-30, 2008, Auckland, New Zealand

idential reasoning [9] or belief function theory, is a mechanism formalized by Shafer [10] for representing and reasoning with uncertain, imprecise and incomplete information. The theory represents a set of propositional hypotheses by a frame of discernment.

Definition 1. **Frames of Discernment** A frame of discernment (or simply a frame), usually denoted as Θ, contains mutually exclusive and exhaustive propositional hypotheses, one and only one of which is true.

For example, a patient has been observed having two symptoms: "coughing" and "sniveling" and only three types of illness could have caused these symptoms: "flu"(F), "cold"(C) and "pneumonia"(P). We can use a frame $\Theta = \{F, C, P\}$ to represent these types of illness.

There are three important functions in DS theory: the Basic Probability Assignment function (*bpa* or m), the Belief function (Bel), and the Plausibility function (Pl).

Definition 2. **Basic Probability Assignment (Mass Function)** A function, m: $2^{\Theta} \rightarrow [0, 1]$, is called a basic probability assignment on a frame Θ if it satisfies the following two conditions:

$$m(\phi) = 0 \tag{1}$$

$$\sum_{A \subseteq \Theta} m(A) = 1 \tag{2}$$

where ϕ is an empty set and A is any subset of Θ.

The Basic Probability Assignment function (mass function thereafter) is a primitive function. Given a frame, Θ, for each source of evidence, a mass function assigns a mass to every subset of Θ, which represents the degree of belief that one of the hypotheses in the subset is true, given the source of evidence. For example, when the patient has been observed having the symptom "coughing", the degree of belief that the patient has "flu" or "cold" is 0.6 and the degree of belief that the patient has "pneumonia" is 0.4. We then have a mass function: $m_1(\{C, F\}) = 0.6$ and $m_1(\{P\}) = 0.4$. Similarly, with the symptom of "sniveling", we have another mass function: $m_2(\{F\}) = 0.7$, $m_2(\{C\}) = 0.2$ and $m_2(\{P\}) = 0.1$.

From the mass function, the Belief function (Bel), and the Plausibility function (Pl) can be derived, which represent the upper and lower bounds of an interval for every subset of Θ, which contains the precise probability that one of the hypotheses in the subset is true, given the source of evidence.

Given two mass functions m_1 and m_2, DS theory also provides Dempster's combination rule for combining them, which is defined as follows:

$$m(C) = \frac{\sum_{A \bigcap B=C} m_1(A) m_2(B)}{1 - \sum_{A \bigcap B=\phi} m_1(A) m_2(B)} \tag{3}$$

In the above example, we combine two mass functions, m_1 and m_2, to get a combined mass function: $m(C) = 0.207$, $m(F) = 0.724$ and $m(P) = 0.069$. Therefore given the two symptoms the patient has, it is more likely that he is having "flu".

3. COMBINING MULTIPLE MATCHERS USING DS THEORY

Based on DS theory, we propose an evidential approach to combining multiple matchers that exploit different features of schemas. Given a source schema and a target schema, for every source attribute, each target attribute is one of its candidate correspondences. An individual matcher provides a different measure on the validity of each candidate correspondence of the source attribute. Applying this measure to all the candidate correspondences provides a source of evidence on the validity of each candidate correspondence. Based on this source of evidence, we can generate a mass function that assigns a mass to every subset of the given frame, reflecting the degree of belief that the valid correspondence of the source attribute belongs to the subset. A set of different matchers provide multiple measures and applying these measures to all the candidate correspondences provides multiple sources of evidence on the validity of each candidate correspondence, based on which we can generate multiple mass functions. These mass functions can then be combined using Dempster's combination rule to decide on the top k attribute correspondences of each source attribute.

3.1 Individual Matchers

We use four individual matchers, the first three matchers are based on different linguistic features of attribute names and the last matcher uses the data types of attributes.

We use WordNet[1], an ontology database, to compute semantic similarity between two words. We use the traditional edge counting approach to measure word similarity. We define semantic similarity between two words, w_1 and w_2, as $Sim_{se}(w_1, w_2) = 1/L$, where L is the shortest path in WordNet between w_1 and w_2.

Edit distance between two strings is measured by the number of edit operations necessary to transform one string into another [11]. We define the edit distance-based string similarity between two words, w_1 and w_2, as follows:

$$Sim_{ed}(w_1, w_2) = \frac{1}{1 + ed(s_1, s_2)} \tag{4}$$

where s_1 and s_2 are two strings in w_1 and w_2 respectively and $ed(s_1, s_2)$ is the edit distance between s_1 and s_2.

Jaro distance between two strings is measured by the number and order of the common characters in them. The Jaro distance-based string similarity $Sim_{ja}(w_1, w_2)$ between two words, w_1 and w_2 is defined as the Jaro distance $Jaro(s_1, s_2)$ [12] between two strings s_1 and s_2 in w_1 and w_2.

Assume that two attribute names, A_1 and A_2, contain two sets of words, $A_1 = \{w_1, w_2, ..., w_m\}$ and $A_2 = \{w'_1, w'_2, ..., w'_n\}$. For each word, w_i for $i = 1, 2, ...m$, in A_1, we calculate its similarity with every word in A_2 and find the maximum similarity value v_i. We then get a similarity value set for A_1: $Sim_1 = \{v_1, v_2, ..., v_m\}$. Similarly, we get a similarity value set for A_2: $Sim_2 = \{v'_1, v'_2, ..., v'_n\}$. We calculate similarity between two attribute names A_1 and A_2 as follows:

$$Sim(A_1, A_2) = \frac{\sum_{i=1}^{m} v_i + \sum_{i=1}^{n} v'_i}{m + n} \tag{5}$$

where m is the number of words in A_1, n is the number of words in A_2.

We define that two data types are compatible if they are the same or one subsumes another (is-a relationship). The similarity value between two attribute names is 1, if their data types are the same. Otherwise it is 0.

[1] http://wordnet.princeton.edu/

3.2 Interpreting Match Results of Individual Matchers

Assume that we have a source schema, $S = \{a_1, a_2, ..., a_m\}$, where a_i, for $i = 1, 2, ..., m$, is a source attribute, and a target schema, $T = \{b_1, b_2, ..., b_n\}$, where b_j, for $j = 1, 2, ..., n$, is a target attribute. For each source attribute, a_i, we have a set of candidate correspondences in the target schema $\{< a_i, b_1 >, < a_i, b_2 >, ..., < a_i, b_n >\}$. It is also possible that a_i may have no correspondence in the target schema at all. We therefore have a frame of discernment for a_i, $\Theta = \{< a_i, b_1 >, < a_i, b_2 >, ..., < a_i, b_n >, < a_i, null >\}$, where $< a_i, null >$ represents that there is no correspondence of a_i in the target schema.

3.2.1 Generating Indistinguishable Subsets of Attribute Correspondences

For some matchers we cluster Θ into a set of indistinguishable subsets. For example, if the data type of a source attribute is compatible with the data types of two candidate correspondences in the target schema, then the two correspondences cannot be distinguished from each other. So we cluster these indistinguishable correspondences into a subset.

3.2.2 Generating Mass Distributions on Indistinguishable Subsets

Given a matcher, for each indistinguishable subset of attribute correspondences, we have a similarity value for each correspondence in the set, which represents how well the two attributes in the correspondence match according to the measure used by the matcher. Suppose the subset is $\{< a_i, b_{i1} >, < a_i, b_{i2} >, ..., < a_i, b_{in} >\}$, a mass assigned to the subset is calculated based on the similarity values for all the attribute correspondences in the subset as follows:

$$m'(A) = 1 - \Pi_{j=1}^{n}(1 - Sim(a_i, b_{ij})) \quad (6)$$

where $Sim(a_i, b_{ij})$ is similarity value for the correspondence $< a_i, b_{ij} >$ in the subset. For the special singleton subset, $\{< a_i, null >\}$, since we do not have a similarity value for it by any matcher, the mass assigned to the subset is calculated as follows:

$$m'(\{< a_i, null >\}) = \Pi_{j=1}^{n}(1 - Sim(a_i, b_{ij})) \quad (7)$$

The mass assigned to $\{< a_i, null >\}$, therefore, represents the degree of belief that none of the target attributes is the attribute correspondence of source attribute, a_i.

We scale the mass distribution, m', by the following formula so that the sum of all masses assigned to every indistinguishable subset equals to 1:

$$m(A) = \frac{m'(A)}{\sum_{B \subseteq \Theta} m'(B)} \quad (8)$$

where A and B are subsets of Θ.

3.3 Combining Mass Functions from Multiple Matchers

We now have a mass function by each of the individual matchers, which assigns a mass to every indistinguishable subset of Θ. Using Dempster's combination rule, we can take into account different sources of evidence witnessed by different matchers by combining the appropriate mass functions by these matchers. The mass function produced after this is used to select the top k attribute correspondences of each source attribute.

4. RESOLVING CONFLICTS BETWEEN ATTRIBUTE CORRESPONDENCES

We have now the top k attribute correspondences of each source attribute, which have been selected for an individual source attribute only. There might be conflicts between attribute correspondences of two source attributes (ie. the best correspondences of two different source attributes are the same target attribute). To resolve any conflicts, the attribute correspondences of source attributes are collectively selected to maximize the sum of all the masses on the attribute correspondence of every source attribute. The algorithm is given in Algorithm 1.

Algorithm 1 Resolving Conflicts

Input: A set of all the possible combinations of attribute correspondences for each source attribute $\Omega = \{C | C = \{< a_1, b'_1 > , < a_2, b'_2 > ... < a_m, b'_m >\}\}$,where $< a_i, b'_i > \in \{< a_i, b_{i1} > , < a_i, b_{i2} >, ..., < a_i, b_{ik} >\}$ (the top k correspondences of a_i)

Output: A collection of attribute correspondences with the maximum sum of the mass values of the correspondences for every source attribute

1: $Max \leftarrow 0; Best \leftarrow null.$
2: **for** each C$\in \Omega$ **do**
3: $Sum = \Sigma_{i=1}^{m} m(< a_i, b'_i >)$, where $m(< a_i, b'_i >)$ is the mass function value of $< a_i, b'_i >$
4: **if** $Sum > Max$ **then**
5: $Max \leftarrow Sum; Best \leftarrow C;$
6: **return** Best

For example, assume that the source schema has three attributes: {*Author*, *Publisher*, *Published Date*}, and the target schema has three attributes: {*Author*, *Keywords*, *Release Date*}. We have the top k ($k = 3$) correspondences of each source attribute as follows:

$\{m(< Author, Author >) = 0.88,$
$m(< Author, null >) = 0.11,$
$m(< Author, Keywords >) = 0.01\},$
$\{m(< Publisher, Author >) = 0.47,$
$m(< Publisher, null >) = 0.40,$
$m(< Publisher, Keywords >) = 0.13\},$
$\{m(< Published\ Date, Release\ Date >) = 0.87,$
$m(< Published\ Date, null >) = 0.13,$
$m(< Published\ Date, Author >) = 0.0\}$

Source attributes *Author* and *Publisher* both have target attribute *Author* as their top correspondence and hence are in conflict. Using Algorithm 1, we get $\{< Author, Author >, < Publisher, null >, < Published\ Date, ReleaseDate >\}$ that has the maximum sum of mass values.

5. EXPERIMENTAL RESULTS

We use a set of 88 query interfaces selected from the ICQ Query Interfaces data set at UIUC, which contains manually extracted schemas of interfaces in 5 different domains, which involve 1:1 matching only (as we have focused on 1:1 matching in this paper). We use three performance metrics: precision, recall, and F-measure. Precision is the percentage of correct matches over all the matches by a matcher. Recall is the percentage of correct matches by a matcher over all

Table 1: Precisions of individual matchers

	Edit distance	Jaro distance	Semantic similarity	Ours
Airfares	83.3%	56.8%	86.4%	92.0%
Autos	84.4%	48.1%	93.1%	96.3%
Books	87.0%	48.8%	92.0%	94.4%
Jobs	68.5%	50.0%	71.0%	91.9%
Estates	86.8%	52.9%	81.6%	93.8%
Average	82.1%	51.3%	84.8%	93.7%

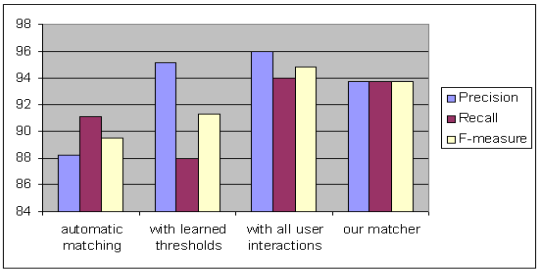

Figure 1: Precision, recall and F-measure of different matchers

the matches by domain experts. F-measure is the incorporation of precision and recall. In our approach, precision, recall and F-measure turn out to be the same. First, in each domain we perform four experiments. We use three individual matchers: edit distance, Jaro distance and semantic similarity (the data type matcher cannot be used alone), and compare their results with our new approach. As shown in Table 1, our matcher gets much higher precision than the individual matchers.

Second, we compare our results with the work in [4], which uses a similar data set for their experiments, covering the same five domains as ours. However, they also handle 1:m matching. In their experiments and a 1:m match is counted as m 1:1 matches. They did three experiments, the first is on automatic matching which uses a weighted strategy to combine multiple matchers and all the thresholds for selecting the combined match results are set to 0. The second uses thresholds learned by user interactions. The last also uses user interactions for resolving uncertainties in match results. As shown in Figure 1, without using learned thresholds, the results of our approach are better. When the learned thresholds are used, their precision is better than ours, but we have higher recall and F-measure. Finally, when user interactions are also used to resolve uncertainties in match results, their results are better than ours. Our approach is effective and accurate for automatic schema matching across query interfaces without automated learning and user interaction.

6. RELATED WORK

Cupid [5] exploits linguistic and structural similarity between elements and uses a weighted formula to combine these two similarities together. However, weights have to be manually generated and are domain dependent.

COMA [7] allows users to tailor match strategies by selecting a combination of match algorithms for a given problem, including Max, Min, Average and Weighted strategies. It also allows users to provide feedback for improving match results. These strategies are effective in some situations while sometimes they cannot combine results effectively, and choosing strategies by users involves human efforts.

In [4], weight coefficients are also used to combine multiple matchers, which are set to some domain-independent empirical values. However, clustering is used to find attribute correspondences across multiple interfaces, in which thresholds are required for merging clusters. These thresholds need to be either manually set or learned from user interactions and are domain dependent. So this approach also involves human effort.

Some approaches [2, 3] use attribute distribution rather than linguistic or domain information. Superior to other schema matching approaches, these approaches can discover synonyms by analyzing attribute distributions in the given schemas. However, they work well only when a large training data set is available, but this is not always the case.

7. CONCLUSIONS

We proposed a new approach to combining multiple matchers using DS theory and presented an algorithm for resolving conflicts among the correspondences of different source attributes. Applying different matchers to a set of candidate correspondences provides different sources of evidence, and mass distributions are defined on the basis of the match results from these matchers. We use Dempster's combination rule to combine these mass distributions, and choose the top k correspondences of each source attribute. Conflicts between the correspondences of different source attributes are finally resolved. We implemented a prototype and tested it using a large data set that contains real-world query interfaces in five different domains. The experimental results demonstrate the feasibility and accuracy of our approach.

8. REFERENCES

[1] Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal. **10**(4) (2001), 334–350.

[2] He, B., Chang, K.C.C.: Statistical schema matching across web query interfaces. SIGMOD'03, 217–228.

[3] He, B., Chang, K.C.C., Han, J.: Discovering complex matchings across web query interfaces: a correlation mining approach. KDD'04, 148–157.

[4] Wu, W., Yu, C.T., Doan, A., Meng, W.: An interactive clustering-based approach to integrating source query interfaces on the deep Web. SIGMOD'04, 95–106.

[5] Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. VLDB'01, 49–58.

[6] Wang, J., Wen, J.R., Lochovsky, F.H., Ma, W.Y.: Instance-based schema matching for web databases by domain-specific query probing. VLDB'04, 408–419.

[7] Do, H.H., Rahm, E.: Coma - a system for flexible combination of schema matching approaches. VLDB'02, 610–621.

[8] Doan, A., Domingos, P., Halevy, A.Y.: Reconciling schemas of disparate data sources: A machine-learning approach. SIGMOD'01, 509–520.

[9] Lowrance, J.D., Garvey, T.D.: Evidential reasoning: An developing concept. ICCS'81, 6–9.

[10] Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press.(1976)

[11] Hall, P., Dowling, G.: Approximate string matching. Computing Surveys. (1980) 381–402.

[12] Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. IIWeb'03, 73–78.