Bootstrapping Uncertainty in Schema Covering

Nguyen Thanh Toan[1], Phan Thanh Cong[1], Duong Chi Thang[2], Nguyen Quoc Viet Hung[3], and Bela Stantic[3]

[1] Bach Khoa University, Ho Chi Minh, Vietnam
[2] Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland
[3] Griffith University, Gold Coast, Australia

Abstract. Schema covering is the process of representing large and complex schemas by easily comprehensible common objects. This task is done by identifying a set of common concepts from a repository called concept repository and generating a cover to describe the schema by the concepts. Traditional schema covering approach has two shortcomings: it does not model the uncertainty in the covering process, and it requires user to state an ambiguity constraint which is hard to define. We remedy this problem by incorporating probabilistic model into schema covering to generate probabilistic schema cover. The integrated probabilities not only enhance the coverage of cover results but also eliminate the need of defining the ambiguity parameter. Experiments on real-datasets show the competitive performance of our approach.

Keywords: Schema matching, Schema covering, Probabilistic models

1 Introduction

Schema matching is the process of finding correspondences between attributes of schemas [1,2]. It is used extensively in many fields [3,4,5], especially data integration [6,7]. Schema matching traditionally performs matching on attribute-level to create attribute correspondences. This process is ineffective considering a large schema with thousands of attributes. Moreover, users tend to think schemas in terms of business object level when designing schema mappings. Therefore, describing the schemas at low-level structure such as attribute makes the manual matching process error-prone. This matching process would be easier if we could represent schemas in a higher level of abstraction.

Since schemas are used to capture everyday business activities and some of these business activities are the same among organizations, these schemas may contain many common parts. These common parts represent business objects that are comprised in the schemas which are called concepts. Some common concepts are “Address”, which describes the location of an entity, or “Contact”, which provides information about a person or an organization. Based on this observation, the process of describing schemas in terms of concepts can be made possible and it is called schema covering. Schema covering is a novel approach which has been studied carefully in [8,9].

In [8], the schema cover found by schema covering must satisfy a pre-defined ambiguity constraint which limits the number of times a schema attribute can be covered.

However, this ambiguity constraint is hard to define since it must be stated beforehand and for each attribute in the schema. Traditional schema covering approach has another shortcoming that it does not support modeling uncertainty arisen in the covering process. As a result, these problems lead to the employment of probability to express uncertainty. We propose incorporating probabilistic model into schema covering to introduce *probabilistic schema covering*.

In short, our goal is to create a new schema covering mechanism that does not require a user-defined ambiguity constraint by incorporating probabilistic model. The paper is organized as follows. In §2, we model and formulate the problem of probabilistic schema cover. In §3, we present the probabilistic schema covering framework. In §4, we run various experiments on probabilistic schema covering, before §5 concludes the paper.

2 Model and Problem Statement

Let schema $s = \{a_1, a_2, \ldots, a_n\}$ be a finite set of attributes. Let s and s' be schemas with n and n' attributes, respectively. Let $S = s \times s'$ be the set of all possible *attribute correspondences* between s and s'. Each attribute correspondence (a pair of attributes) is associated with a confidence value $m_{i,j}(s, s') \in [0, 1]$ which represents the similarity between the i-th attribute of s and the j-th attribute of s' [10,11].

A concept c is also a set of attributes: $c = a_1, a_2, ..., a_m$ where a_i is an attribute. A concept and a schema is basically the same as they are both sets of attributes. However, a concept is more meaningful as it describes a business object and it also has a smaller size. Concepts have relations between them called *micromappings*. Each micromapping is actually a set of attribute correspondences. We also define the counterpart of concepts in the schema which are subschemas. A subschema t is also a set of attributes and it is a subset of schema s. Each concept and its subschema has an alignment score $f(t, c)$ which describes the similarity between them.

In general, the schema covering framework mentioned in [8] takes a schema and a prebuilt concept repository as input. The concept repository is a corpus of predefined concepts, which is built before-hand [8].

Definition 1. *Given a set of subschemas T_s of schema s, a set C of concepts, we define a set of valid matchings between subschemas and concepts:*

$$E(T_s, C) = \{(t, c) | t \in T_s, c \in C\}$$

where (t, c) is a set of attribute correspondences between subschema t and concept c. A cover of s by C, $v_{s,C} \subseteq E(T_s, C)$ is a subset of valid matchings between T_s and C.

The schema cover found by traditional schema covering approach must satisfy an ambiguity constraint which limits the number of times a schema attribute can be covered. Therefore, traditional schema covering approach is also called ambiguity-based schema covering, which is discussed in [8]. Having described the traditional schema covering approach, we can turn to the problem we want to solve.

Formally, our problem takes a set of $\langle subschema, concept \rangle$ pairs, $E(T_s, C) = \{(t, c) | t \in T_s, c \in C\}$, as input where T_s is a set of sub-schemas and C is a set of

concepts in the repository. Each pair is attached with an alignment score $f(t, c)$ where f is a user-defined function. In this problem, we want to compute a probabilistic schema cover. It is a set of possible covers v_i and each cover is associated with a probability $Pr(v_i)$. The formal definition for probabilistic schema cover is described as follows.

Problem 1 (Probabilistic Schema Cover)**.** *Let E be a set of $\langle subschema, concept \rangle$ pairs. The probabilistic schema cover built from E is a set $V = \{(v_1, Pr(v_1)), \ldots, (v_n, Pr(v_n))\}$ such that*

- *For each $i \in [1, n]$, v_i is a cover and for every $i, j \in [1, n]$, $i \neq j \Rightarrow v_i \neq v_j$*
- *$Pr(v_i) \in [0, 1]$ and $\sum_{i=1}^{n} Pr(v_i) = 1$*

3 Probabilistic schema covering

The probabilistic schema covering framework has three steps as described in Fig. 1. It takes a set of pairs after decomposition E as input and return a probabilistic schema cover containing a set of covers with probabilities attached to each of them.

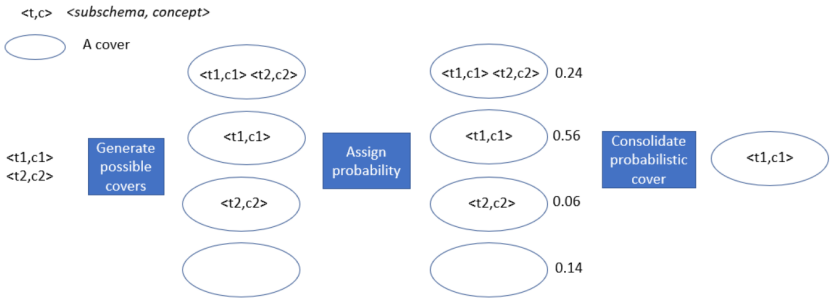

Fig. 1. The probabilistic schema covering framework

3.1 Generate all possible covers

From a set of pairs $E = \{(t, c) | t \in T_s, c \in C\}$ found after decomposing the schema, we generate all its subsets $\Omega = \{v_i | v_i \subset E\}$. Generating its subsets using all the pairs would lead to computational explosion since the size of Ω, $|\Omega| = 2^{|E|}$, is large. Therefore, we need some methods to reduce the computational space.

We introduce the alignment score threshold λ and the error window ϵ to decrease the size of the computational space. Using the threshold λ and the error window ϵ, we define two sets of pairs E_c and E_u:

- Certain set $E_c = \{(t, c) \in E | f(t, c) \geq \lambda + \epsilon\}$
- Uncertain set $E_u = \{(t, c) \in E | f(t, c) < \lambda + \epsilon \wedge f(t, c) \geq \lambda - \epsilon\}$

By setting the alignment score threshold λ, we want to focus only on the promising pairs. Pairs with alignment scores higher than the threshold are more likely to be correct. On the other hand, the error window value ϵ represents pairs that we are unsure if they are correct or not. That means we need to assign probabilities to only these pairs in E_u to express uncertainty.

From the uncertain set of pairs E_u, we generate the possible covers $\Omega_u = v_i^* | v_i^* \subset E_u$. Therefore, the number of possible covers $|\Omega_u|$ is $2^{|E_u|}$. Since $2^{|E_u|} \ll 2^{|E|}$, we have reduced the computational space significantly. Finally, the probabilistic schema cover for E is computed based on Ω_u as follows: $\Omega = \{v_i | v_i = v_i^* \cup E_c, v_i^* \in \Omega_u\}$ and $Pr(v_i) = Pr(v_i^*)$.

3.2 Assign probability to each cover

After the first step, we have generated a set of possible covers Ω_u from the uncertain set of pairs E_u. In this step, we assign probability to each cover $v_i^* \in \Omega_u$.

Consistency constraint. Despite the fact that alignment scores express how similar between the subschemas and the concepts, they do not tell us which concept a subschema should align to.

Definition 2. *A probabilistic cover V is consistent with a pair (t, c) if the sum of probabilities of all covers that contain (t, c) equals the alignment score $f(t, c)$. A probabilistic cover V is consistent with a pair (t, c) if*

$$\sum_{(t,c) \in v_i} Pr(v_i) = f(t, c)$$

A probabilistic cover V is consistent with a set of pairs M if it is consistent with each pair in M.

This constraint is introduced to ensure that a cover containing a pair with low alignment score has low probability. Since a pair with low alignment score is more likely to be incorrect, the cover in which it participates is also less likely to be correct.

Entropy maximization. The probability assignment problem can now be reformulated to a constraint optimization problem (OPT). That is, we need to assign the probabilities to the covers in a probabilistic cover such that both the consistency constraint is satisfied and the entropy is maximized. The optimization problem is described as follows.

Definition 3. *Let $Pr(v_1), \ldots, Pr(v_n)$ be the probabilities of cover $v_1, \ldots, v_n$ respectively. $Pr(v_i)$ is found by solving the following OPT problem:*

maximize $\sum_{i=1}^{n} -Pr(v_i) \log Pr(v_i)$, subject to:

1. $\forall i \in [1, n], 0 \leq Pr(v_i) \leq 1$
2. $\sum_{i=1..n} Pr(v_i) = 1$
3. $\forall (t, c) \in E_u : \sum_{j \in [1,n]:(t,c) \in v_j} Pr(v_j) = f(t, c)$

4 Experiments

Dataset. We start by introducing the dataset being used for evaluation. In fact, finding an appropriate dataset is a non-trivial task as the collected schemas must be relevant and belong to a same domain. We have collected 5 schemas from the Purchase Order domain. Their statistics are described in Table 1. From these schemas, we also create

Table 1. Statistics of the five schemas

	Apertum	CIDX	Excel	Noris	Paragon
#Nodes	140	40	54	65	77
#Internal Nodes	25/115	7/33	12/42	11/54	12/65
Depth	4	3	3	3	5

Table 2. #Golden mappings between schemas

	Apertum	CIDX	Excel	Noris	Paragon
Apertum		54	79	85	66
CIDX	54		65	32	49
Excel	79	65		50	60
Noris	85	32	50		45
Paragon	66	49	60	45	

the golden mappings between them manually. The number of goldenmappings between pairs of schemas is described in Table 2.

Concept repository. We build the concept repository by COMA++ [12] with default parameters, resulting in 45 concepts, 50 micromappings, 220 attributes, 5.089 attributes per concept in average, 1.11 micromappings per concept in average.

Metrics. Let R be the set of correct correspondences found manually. Let F denote the set of correspondences that we generate (or we consider them to be correct). Let $I = R \cap F$ denote the actual correct correspondences in F. In order to evaluate the result, we use two typical metrics: precision, which is $|I|/|F|$ and recall, which is $|I|/|R|$. A high value of both precision and recall is desired. As it is hard to find the correct cover from such a large concept repository, we take a different approach to calculate the precision and recall. For each subschema and concept pair, we calculate its precision and recall value then we take the average to get the precision, recall of the whole cover.

Effects of score threshold and error window on cover result. In this experiment, we want to find the cover of schema Excel by the concept repository. We vary the threshold, error window to see their effects on the final cover. To consolidate cover, we select the cover with the highest probability. The result is shown in Fig. 2. In general, precision and recall are high, both of them are higher than 60%. This means that we can find a good cover. Intuitively, the precision and recall increase when the threshold are higher. This is reasonable that the higher the threshold is, we only consider the more likely-correct pairs.

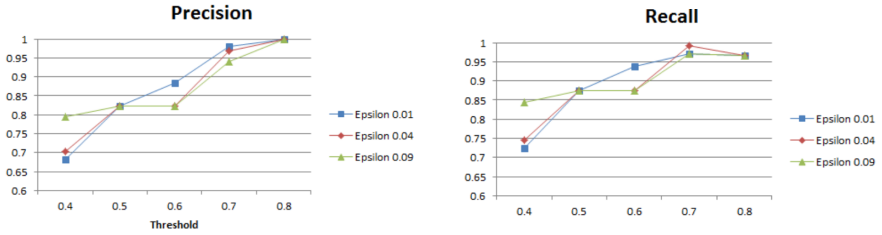

Fig. 2. Precision and recall of the cover of schema Excel

A comparison with ambiguity-based schema covering. In this experiment, we compare probabilistic schema covering with the ambiguity-based schema covering approach mentioned in [8]. Fig. 3 shows the comparison of two approaches on the precision and recall value. With a low threshold, ambiguity-based covering has higher precision and recall. However, as we analyze the cover chosen by ambiguity-based covering, we found that this cover contains no pair that has alignment score lower than 0.5. On the other hand, probabilistic covering also consider various pairs with low alignment score that results in lower precision and recall.

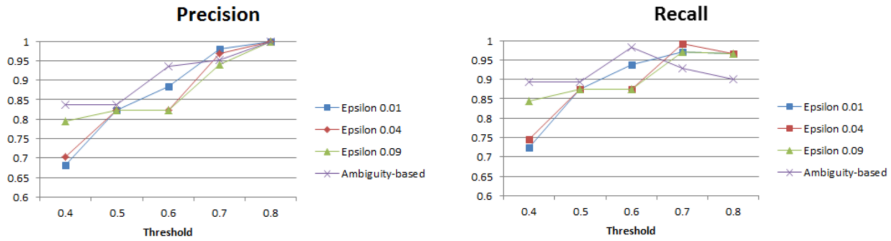

Fig. 3. A comparison with ambiguity-based covering

5 Conclusions

This paper describes a novel approach to schema covering in order to mitigate uncertainty and improve covering results: probabilistic schema covering. In order to propose this approach, we have solved the problem of finding a mechanism to integrate probabilistic model into schema covering In order to generate a probabilistic schema cover, we first construct its possible set of covers and then we assign probability to each cover. The assigned probabilities must satisfy a consistency constraint and their entropy must also be maximized. Throughout the experiments, we have shown that probabilistic schema covering is a robust approach and competitive to traditional schema covering approach.

References

1. Hung, N.Q.V., Luong, X.H., Miklós, Z., Quan, T.T., Aberer, K.: Collaborative schema matching reconciliation. In: CoopIS. (2013) 222–240
2. Hung, N.Q.V., Tam, N.T., Chau, V.T., Wijaya, T.K., Miklós, Z., Aberer, K., Gal, A., Weidlich, M.: SMART: A tool for analyzing and reconciling schema matching networks. In: ICDE. (2015) 1488–1491
3. Hung, N.Q.V., Tam, N.T., Miklós, Z., Aberer, K.: On leveraging crowdsourcing techniques for schema matching networks. In: DASFAA. (2013) 139–154
4. Hung, N.Q.V., Luong, X.H., Miklós, Z., Quan, T.T., Aberer, K.: An MAS negotiation support tool for schema matching. In: AAMAS. (2013) 1391–1392
5. Hung, N.Q.V., Tam, N.T., Miklós, Z., Aberer, K.: Reconciling schema matching networks through crowdsourcing. EAI (2014) e2
6. NGUYEN, Q.V.H.: Reconciling Schema Matching Networks. PhD thesis, Ecole Polytechnique Federale de Lausanne (2014)
7. Gal, A., Sagi, T., Weidlich, M., Levy, E., Shafran, V., Miklós, Z., Hung, N.Q.V.: Making sense of top-k matchings: A unified match graph for schema matching. In: IIWeb. (2012) 6
8. Saha, B., Stanoi, I., Clarkson, K.L.: Schema covering: a step towards enabling reuse in information integration. In: ICDE. (2010) 285–296
9. Gal, A., Katz, M., Sagi, T., Weidlich, M., Aberer, K., Hung, N.Q.V., Miklós, Z., Levy, E., Shafran, V.: Completeness and ambiguity of schema cover. In: CoopIS. (2013) 241–258
10. Hung, N.Q.V., Wijaya, T.K., Miklós, Z., Aberer, K., Levy, E., Shafran, V., Gal, A., Weidlich, M.: Minimizing human effort in reconciling match networks. In: ER. (2013) 212–226
11. Hung, N.Q.V., Tam, N.T., Miklós, Z., Aberer, K., Gal, A., Weidlich, M.: Pay-as-you-go reconciliation in schema matching networks. In: ICDE. (2014) 220–231
12. Arnold, P., Rahm, E.: Enriching ontology mappings with semantic relations. Data & Knowledge Engineering **93** (2014) 1–18