Ontology Mapping enhanced using Bayesian Networks

Ondřej Šváb, Vojtěch Svátek

University of Economics, Prague, Department of Information and Knowledge Engineering,
Winston Churchill Sq. 4, 130 67 Praha 3, Prague, Czech Republic
svabo@vse.cz, svatek@vse.cz

Abstract. Bayesian networks (BNs) can capture interdependencies among ontology mapping methods and thus possibly improve the way they are combined. We outline the basic idea behind our approach and show some experiments on ontologies from the OAEI 'conference organisation' collection. The possibility of modelling explicit mapping patterns in combination with methods is also discussed[1].

Keywords: Semantic Web, Ontology Mapping, Bayesian Networks, Patterns

1 Introduction

Ontologies can help integrate semantic views on real-world data. Unfortunately, designers of ontologies themselves apply different views of the same domain during ontology development. This yields semantic heterogeneity at ontology level, which is one of main obstacles to semantic interoperability. *Ontology mapping* (also called 'matching' or 'alignment') is the core component of approaches attempting to solve this problem. It consists in finding mappings (also called 'correspondences') among entities (classes, relations) from different ontologies. The set of mappings is called alignment. The process of mapping is followed by ontology merging, ontology transformation, data transformation etc. A survey of ontology mapping methods is in [9].

As a rule of thumb, most existing systems for ontology mapping combine various methods for achieving higher performance in terms of recall and precision. Our approach relies on *Bayesian networks* (BNs) as well-known formal technique that can capture interdependencies among random variables. We believe that this approach can bring additional benefits compared to ad hoc combination of methods, mainly resulting from better adaptability (training from data within a well-established formal framework).

The paper is structured as follows. Section 2 very briefly explaines the notion of Bayesian networks. Section 3 reports on previously published research projects

[1] This paper is a significantly extended version of a paper presented as poster at the Ontology Matching workshop (International Workshop on Ontology Matching, OM-2006 at ISWC-2006 held in Athens, Georgia, USA).

that applied the BN technology to ontology mapping. Section 4 presents the basic principles of our approach and the way low-level mapping methods and mapping patterns can be captured in the BN model. Preliminary experiments on combining low-level string-based methods are the subject of section 5. Section 6, in turn, elaborates on the problem of mapping pattern modelling. Finally, section 7 summarises the paper and outlines directions for future research.

2 Bayesian Networks

A Bayesian network (BN) [6] is a directed acyclic graph with attached local probability distributions. Nodes in the graph represent random variables (corresponding to attributes, features etc.). Each random variable has a mutually exclusive and exhaustive set of values (states). Edges in the graph represents direct interdependences between two random variables.

Bayesian networks consist of two sort of knowledge:

- qualitative knowledge that describes interdependencies by means of *directed graph*
- quantitative knowledge that captures relations among random variables by means of *conditional probability tables* (CPTs)

An advantage of BNs compared to other uncertainty representation formalisms is the possibility to model complicated mutually related phenomena in quite a tractable way.

3 Prior Research on Using BNs for Ontology Mapping

Two approaches that use BNs for Ontology Mapping have recently been reported.

The first one is *OMEN* (Ontology Mapping Enhancer, [7]), which mainly serves for enhancing existing mappings. Its input are results of another mapping tool, while its output are more precise mappings as well as and new mappings. Nodes in the BN represent pairs of concepts that can potentially be mapped. Edges follow the taxonomy given in original ontologies (non-taxonomic relationships are not considered). The network structure thus closely mimics that of ontologies themselves, though several heuristics for graph pruning are employed in this transformation. For constructing conditional probability tables (CPTs) for each node so-called meta-rules are used; an example of (basic) meta-rule is: "if two nodes in an ontology graph match and so do two arrows coming out of these nodes then the probability that nodes at the other end of the arrows match as well is increased". Prior probabilities, as well as the evidence for propagating evidence through BN and thus enhancing existing mappings and getting new ones, is obtained from the output of another mapping tool.

The second relevant project, *BayesOWL* ([8]), is rather an underlying framework for ontology mapping than a mapping method per se. BayesOWL is originally an extension of the popular ontology language OWL in terms of capturing the probability associated with domain entities. The probabilistic ontological information is assumed to be learnt (in forms of probabilistic constraints) from web data using a text-classification-based learner; this information is translated to BNs. Mappings among concepts from two different ontologies then can be discovered using so-called evidential reasoning across two BNs.

Finally, let us remark that inductively trained models have been also been used for ontology mapping beyond the Bayesian setting; for example, Ehrig et al. [4] used classification trees, neural nets and support-vector machines.

4 Modelling Dependencies among Mapping Methods

4.1 Basic Principles

Our approach described in this paper differs from prior approaches in the sense that we don't apply BN modelling to ontologies or their mappings themselves but rather to different *mapping methods*. The BNs considered are assumed to contain nodes[2] representing the results of individual methods plus one representing the final output. This will allow us not only to combine the methods (in the probabilistic framework) but also to talk about conditionally in/dependent methods, a minimal required subset of methods and the like.

The mapping methods can have varying degree of granularity; low-level methods, understood as *mapping justifications*, are the main focus in our completed experiments. Moreover, in the work-in-progress part of our research, we account for *mapping patterns* encompassing small structural fragments of ontologies. The patterns will capture, to some degree, similar information as OMEN meta-rules, we however prefer to model them explicitly within the BN formalism rather using a separate formalism. We will discuss the issues of low-level methods and mapping patterns in the next two subsections.

4.2 Focus on Low-Level Methods

When dealing with different mapping methods, we can distinguish among *families of methods* (such as string-based, linguistic-resources-based, graph-based, logic-based etc.) sharing some generic principle and input resources. Each such family then encompasses multiple 'low-level methods'; for example, a string-based method can be built upon diverse string distance measures or a combination of them. We will call each low-level technique *mapping justification*. As families of methods can still be internally rather heterogeneous, we believe that low-level techniques are a more meaningful target for BN modelling, as their statistical dependencies are likely to reflect plausible relationships even interpretable by human. We thus decided to dedicate a separate node of the BN to each mapping justification.

[2] Or, possibly, sub-networks, when considering structural patterns, see later.

Fig. 1. The simplest mapping pattern across two ontologies

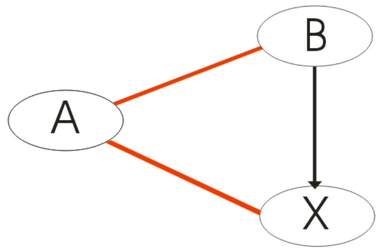

Fig. 2. Example of mapping pattern across two ontologies

4.3 Use of Mapping Patterns

Reusable patterns are currently a hot topic in ontological engineering [1]. The notion of *mapping pattern* is a natural counterpart to that of intra-ontology ('design') pattern, in the same way as a (formalised) mapping language is to an ontology representation language (such as OWL). Mapping patterns have been e.g. implicitly proposed by Ghidini & Serafini [5], who even consider mappings among different modelling constructs (such as concept-to-relation). A mapping pattern is, essentially, a structure[3] containing some (at least one) constructs from each of the two (or more) ontologies plus some (candidate) mapping among them.

The simplest mapping pattern only considers one concept from each of the two ontologies (see Figure 1). An example of a bit more complex mapping pattern is depicted in Figure 2. The left-hand side (class A) is from O_1 and the right-hand side (class B and its subclass X) is from O_2. Here we try to map class A from to class B and simultaneously we try to map class A to class X. This mapping pattern will be referred to in section 6, where we discuss the way of reflecting mapping patterns in the BN structure.

4.4 Training the BN

The input to the process of BN training for ontology mapping are positive and negative examples with results of individual methods ('mapping justifications'), and possibly also the network structure, unless we want to learn it as well. The

[3] Note that we don't consider other aspects of *design* patterns, such as motivations for their usage; the acts in ontology mapping, unlike those in ontology design, are typically enforced by already present ontological heterogeneity and hence are less subject to human deliberation.

positive examples correspond to pairs for which mapping has previously been established, while the negative ones are (all or a subset of) pairs that have been identified as non-matching. Then CPTs and possibly the structure are learnt. In the phase of using the trained BN, the mapping justifications for unseen cases (pairs of concepts) are counted and inserted into the BN as evidence. The result of alignment is calculated via propagation of this evidence.

Especially when considering patterns, the effectivity of automated, training-based mapping seems to strongly depend on the nature of available data. Let us now consider two scenarios[4]: training on a sample of current data (ontology pair or collection) and training on other data. The first scenario can be viewed as situation in which two ontologies O_1, O_2 and a partial alignment A are given. Our task is to construct a more comprehensive alignment $A' \supset A$. The second scenario can be viewed as situation in which we have no mappings. It means that we have to construct the alignment B from scratch. Let us now formulate some preliminary (partly obvious, partly speculative) hypotheses on these two scenarios.

If we train the BN on sample mappings from the same ontology pair we want to align in the end, mapping methods with high quality on training data are likely to perform well on new data as well. This however depends on whether each of the ontologies is homogeneous in terms of design patterns; naming conventions are particularly influential when applying string-based methods.

On the other hand, if we train the BN on heterogeneous/extraneous data then the performance of different methods becomes less predictable. Still, even if the new ontologies to be mapped are different from the training ones, the portability of the trained BN will probably be higher if they belong to the same domain of concern, as they could reflect the design patterns (incl. naming conventions) of people from a certain community. However, mutual in/dependencies among methods (cf. section 5), may emerge even then, as they would typically only depend on the methods and not on particular ontologies or possibly even domains.

5 Experiments

For experiments we choose ontologies from the *OntoFarm* collection [11], which is currently part of the OAEI 2006 setting[5]. This collection models the domain of conference organisation; individual ontologies have been designed quite independently (by different people with no or little contacts among themselves) based on different resources: personal experience with conference organisation, conference web pages or conference organisation support tools.

For simplicity, we restricted the first experiments to string distance measures as mapping justifications. In particular, we used the implementation of several

[4] The issue of generic process of alignment generation with training phase has been more thoroughly elaborated in the APFEL project [4].

[5] *OntoFarm* can be downloaded from `http://nb.vse.cz/~svabo/oaei2006/`

measures from the *SecondString* library[6], namely the following ten ones: Levenshtein, Jaro, Jaccard, Char-Jaccard, Smith-Waterman, Monge-Elkan, SLIM, TokenFelligiSunter, UnSmoothedJS and TFIDF. Because of the local nature of distance string measures, capturing context by means of mapping patterns does not seem to bring great benefits; we thus only focused on the combination of low-level methods.

The experiments were conducted in order to answer following question: *Do we get any benefit from the combination (using BNs) of string distance measures?*

We extracted classes from two ontologies [7]. Our training data consist of 798 pairs, of which 149 were manually labelled as positives and 649 as negatives. They were 'semi-randomly' picked from different parts of the ontology; the overall number of possible pairs would be about 2500 (the product of numbers of concepts in both ontologies). The mapping justification of each method were transformed from 0 to 1 scale to two categories 'true' if the value is over 0.5 and 'false' if the value is lower or equal to 0.5.

To learn the BN (both the structure and CPTs) we use the Hugin tool[8]. The structure was was trained using the NPC method [10]. The level of significance was at 0.05. CPTs were trained using the EM algorithm [3][9].

We learnt two Bayesian networks in this way. The first one has been enforced the *naive Bayesian structure*, which assumes independence of methods, as seen at Fig. 3. Only the CPTs were learned from data. For the second network (see Figure 4), we also *learnt the structure*; in this way we could also explore interdependencies among low-level methods. From the structure of BN and the defintion of so-called *Markov blanket* [6] we can conclude that if we know the mapping justifications of TFIDF, Smith-Waterman, Jaccard, Jaro, and SLIM, other methods do not matter. Also we include only such methods that were interrelated to some other method. This is the reason why TokenFelligiSunter and UnSmoothedJS measures are not in the BN at all. The interpretation of this BN would be as follows: given partly mapped concepts between two ontologies, it shows that for the next phase we can do with those five mapping justifications (TFIDF, Smith-Waterman, Jaccard, Jaro and SLIM) only.

To evaluate the performance of each proposed Bayesian classifier (in terms of *precision*, *recall* and *accuracy*) we used the *one-leave-out method* over our training data. During the tracing of performance we experimented with probability thresholds. For the naive Bayesian classifier, we got the best result with threshold 80%: 73% precision, 60% recall (F-measure was then 0.66) and 88% accuracy. For the Bayesian classifier with learnt structure we got 84% precision, 53% recall (F-measure was 0.65) and 89% accuracy as best result, for whatever threshold between 40% and 70%.

[6] `http://secondstring.sourceforge.net/`

[7] Namely, `ekaw.owl` and `ConfOf.owl` from the mentioned collection

[8] `http://www.hugin.com/`

[9] As there were no missing data in our training data set, it only had to be run in one cycle.

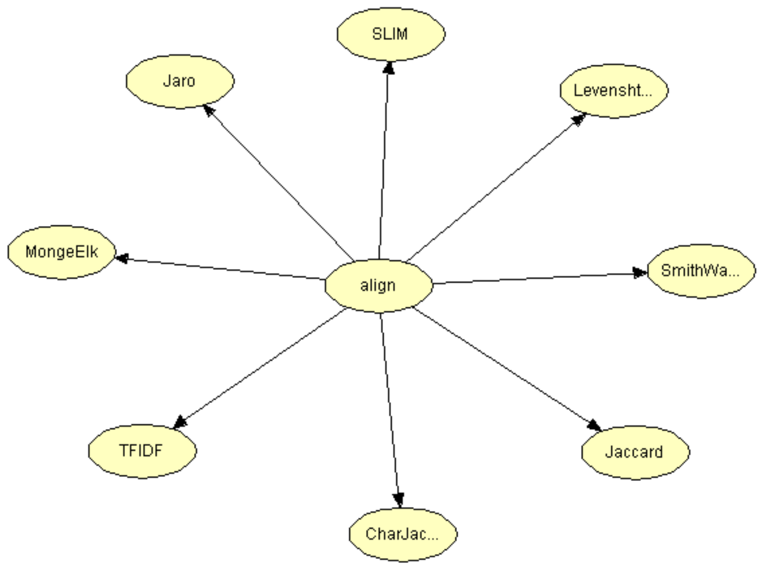

Fig. 3. BN - naive Bayes structure

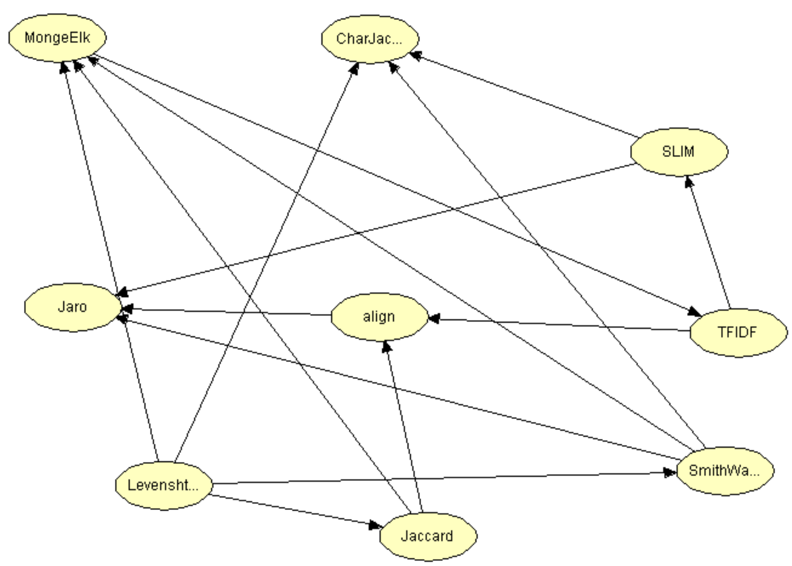

Fig. 4. BN - automatically learnt structure

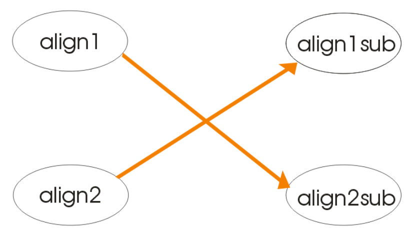

Fig. 5. Fragment of BN reflecting the mapping pattern from Fig. 2

Both our classifiers outperform trivial classifiers that always predict true or false, respectively. Overall, the Bayesian classifier with learnt structure outperformed the naive Bayesian classifier. On the other hand, the best individual method (Jaccard) performed the same as the Bayesian classifier with learnt structure (84% precision, 53% of recall and 89% accuracy) with threshold around 50%. According to this result, we could say that the combination (using BN) of string distance measures does not bring a direct benefit. However, the (second) Bayesian classifier is less sensitive to the change of threshold, as Jaccard moves towards 100% precision but rather low recall of 23% as soon as the threshold increases to 60%.

6 Towards a Pattern-Based Bayesian Model

In order to reflect mapping patterns such as that from Fig. 2, we have to redesign the structure of the BN. Rather than with a single node, each method (and the final result) would be represented with a set of nodes reflecting the given pattern. For example, a fragment of BN reflecting the mentioned mapping pattern is depicted in Fig. 5. In this case, we consider not only the equivalence relation but also the (proper) subsumption relation. This fragment of BN has four nodes that represent the alignment of each pair and each relation (equivalence of A and B, equivalence of A and X, subsumption of A and B and subsumption of A and X). `align1` represents the equivalence mapping between A and B. `align1sub` represents the subsumption mapping between A a B ($B \supset A$). `align2` represents the equivalence mapping between A a X. Finally, `align2sub` represents the subsumption mapping between A a X ($A \supset X$). Edges should appear between the pairs of nodes `align1` and `align2sub`, and `align2` and `align1sub`, respectively, due to strict dependencies: if B is equivalent to A then B's subclass X should be subsumed by A, and if A is equivalent to X then X's superclass B should subsume A. The number of nodes representing one low-level method will be twice as high (for equivalence and subsumption, respectively) as the number of candidate mapping edges in the mapping pattern.

7 Conclusions and Future Work

In this paper we described our idea of using low-level methods as mapping justifications in order to train a Bayesian network on a sample of mappings, which would in turn yield new mappings. We presented preliminary results of our experiments with string distance measures as low-level methods that are easiest available in the form of implementations. The results are not entirely convincing in terms of improved performance, which can possibly be explained by too strong correlation among the string-based methods used; this correlation was actually discovered when learning the BN structure in the second experiment. However, the main role of this initial phase of research was to provide us deeper insight into the problems addressed. The possibility to model explicit mapping patterns in combination with methods was also studied; it has not been reflected in experiments to date, although we see it as one of possible ways to overcome the limitations of the previous experiments.

Let us summarise the main directions of our future work. First of all, we would like to focus on capturing the surrounding *context* of mapped concepts. One step in this direction is to concentrate on determining useful mapping patterns and building appropriate BNs reflecting them. Another, partially independent step is to employ, in the role of mapping justifications, not only string-based (low-level) techniques, but also graph-based techniques of mapping and/or lexical (e.g. thesauri-based) techniques. In addition, we would like to implement simple, transparent and *less correlated* string-based justifications (in addition to current, often sophisticated but largely correlated ones) that would correspond to simple reasons why two concepts should be aligned, e.g. identical names, identical prefix, identical suffix, identical infix etc. Thus, we hopefully get string-based mapping justifications where one would benefit from other.

Another interesting aspect is the fact that concepts usually belong to *more than one mapping pattern*. Even in this case, we could consider using BN for combination of decisions from particular BNs dealing with certain mapping patterns.

We also plan to explore whether a more expressive probabilistic representation than traditional BNs could help. An example is first-order Bayesian logic, particularly the Multi-Entity Bayesian Networks formalism (MEBN, [2]), which enables to express a varying number of entities, different entity types and repeated patterns.

Acknowledgement

We thank Jiří Vomlel for his assistance with Bayesian Networks. The research leading to this paper was partially supported by the IGA VSE grant no.26/05 "Methods and tools for ontological engineering", by the IGA VSE grant no.12/06 "Integration of approaches to ontological engineering: design patterns, mapping and mining", and by the Knowledge Web Network of Excellence (IST FP6-507482).

References

1. W3C Semantic Web Best Practices and Deployment Working Group. Ontology Engineering and Patterns Task Force (OEP). Online at `http://www.w3.org/2001/sw/BestPractices/OEP/`
2. Costa, P. C. G., Laskey K. B.: Multi-Entity Bayesian Networks without Multi-Tears. USA 2005 (draft, online `http://ite.gmu.edu/~klaskey/papers/Costa_Laskey_MEBN_wo_Tears.pdf`)
3. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, 39(1):138. 1977.
4. Ehrig M., Staab S., Sure Y.: Bootstrapping Ontology Alignment Methods with APFEL In: Proceedings of ISWC, Galway, Ireland, 2005.
5. Ghidini C., Serafini L.: Reconciling concepts and relations in heterogeneous ontologies. In: Proc. 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 2006.
6. Jensen F. V.: Bayesian Networks and Decision Graphs. Springer, 2001.
7. Mitra P., Noy N. F., Jaiswal A. R.: OMEN: A Probabilistic Ontology Mapping Tool. In: Workshop on Meaning coordination and negotiation at the Third International Semantic Web Conference (ISWC-2004), Hiroshima, Japan, 2004.
8. Pan R., Ding Z., Yu Y., Peng Y.: A Bayesian Network Approach to Ontology Mapping. In: Proceedings ISWC 2005.
9. Shvaiko P., Euzenat J.: A Survey of Schema-based Matching Approaches. *Journal on Data Semantics*, 2005.
10. Spirtes P., Glymour C., Scheines R.: Causation, Prediction, and Search. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Massachusetts, second edition, 2000.
11. Šváb O., Svátek V., Berka P., Rak D., Tomášek P.: OntoFarm: Towards an Experimental Collection of Parallel Ontologies. Poster Session at ISWC 2005.