Supervised Learning of an Ontology Alignment Process

Marc Ehrig[1], Steffen Staab[2], and York Sure[1]

[1]Institute AIFB, University of Karlsruhe
[2]ISWeb, University of Koblenz-Landau

1 Introduction

Semantic alignment between ontologies is a necessary precondition to establish interoperability between agents or services using different ontologies. Thus, in recent years different methods for automatic ontology alignment have been proposed to deal with this challenge. Thereby, the proposed methods were constricted to one of two different paradigms: Either, *(i)*, proposals would include a manually predefined automatic method for proposing alignments, which would be used in the actual alignment process (cf. [4, 5, 8]). They typically consist of a number of substrategies such as finding similar labels. Or, *(ii)*, proposals would learn an automatic alignment method based on instance representations, e.g. bag-of-word models of documents (cf. [3]). Both paradigms suffer from drawbacks. The first paradigm suffers from the problem that it is impossible, even for an expert knowledge engineer, to predict what strategy of aligning entities is most successful for a given pair of ontologies. This is especially the case with increasing complexity of ontology languages or increasing amounts of domain specific conventions. The second paradigm is often hurt by the lack of instances or instance descriptions. Also, knowledge encoded in the intensional descriptions of concepts and relations is only marginally exploited by this way.

Hence, there remains the need to automatically combine multiple diverse and complementary alignment strategies of *all* indicators, i.e. extensional *and* intensional descriptions, in order to produce comprehensive, effective and efficient semi-automatic alignment methods. Such methods need to be flexible to cope with different strategies for various application scenarios, e.g. by using parameters. We call them "Parameterizable Alignment Methods" (PAM). We have developed a bootstrapping approach for acquiring the parameters that drive such a PAM. We call our approach APFEL for "Alignment Process Feature Estimation and Learning".

2 Foundations

2.1 Ontology

In the understanding of this paper an ontology consists of both schema and instantiating data. An ontology O is therefore defined through the following tuple: $O := (C, H_C, R_C, H_R, I, R_I, A)$. Concepts C of the schema are arranged in a subsumption hierarchy H_C. Relations R_C exist between pairs of concepts. Relations can also be arranged in a hierarchy H_R. (Meta-)Data is constituted by instances I of specific concepts. Theses instances are interconnected by relational instances R_I. Additionally one can define axioms A which can be used to infer knowledge from already existing knowledge. Common languages to represent ontologies are RDF(S)[1] or OWL[2], though one should note that each language offers different modeling primitives.

2.2 Alignment

We here define our use of the term "alignment" similarly to [7]: Given two arbitrary ontologies O_1 and O_2, aligning one ontology with another means that for each entity $e \in \mathcal{E}$ (concept C, relation R_C, or instance I) in ontology O_1, we try to find a corresponding entity, which has the same intended meaning, in ontology O_2. The result are alignments between pairs of entities of the two ontologies. Semantically the alignment returns two entities linked by an identity relation.

2.3 General Alignment Process

APFEL is based on the general observation that alignment methods like QOM [4] or PROMPT [8] may be mapped onto a generic alignment process. Major steps include:

1. Feature Engineering, i.e. select small excerpts of the overall ontology definition to describe a specific entity (e.g., the `label` to describe the concept `o1:Daimler`).
2. Search Step Selection, i.e. choose two entities from the two ontologies to compare (e.g., `o1:Daimler` and `o2:Mercedes`).
3. Similarity Assessment, i.e. indicate a similarity for a given description of two entities (e.g., $\mathrm{simil}_{\mathrm{superConcept}}$(`o1:Daimler` ,`o2:Mercedes`)=1.0).
4. Similarity Aggregation, i.e. aggregate multiple similarity assessment for one pair of entities into a single measure (e.g., simil(`o1:Daimler` ,`o2:Mercedes`)=0.5).
5. Interpretation, i.e. use all aggregated numbers, a threshold and interpretation strategy to propose the alignment (align(`o1:Daimler`)='$\perp$').
6. Iteration, i.e. as the similarity of one entity pair influences the similarity of neighboring entity pairs, the equality is propagated through the ontologies (e.g., it may lead to a new simil(`o1:Daimler` ,`o2:Mercedes`)=0.85, subsequently resulting in align(`o1:Daimler`)=`o2:Mercedes`).

3 APFEL

In this section the APFEL process is explained in detail following the Figure 1. Data structures are illustrated through white boxes and process steps through colored boxes.

3.1 Data Structures

APFEL requires two ontologies O_1 and O_2 as inputs to its processing. Either these are the ontologies for which the further alignment process will be optimized directly. Or, they exemplarily represent a type which requires an optimized alignment method.

[1] http://www.w3.org/TR/rdf-schema/
[2] http://www.w3.org/TR/owl-guide/

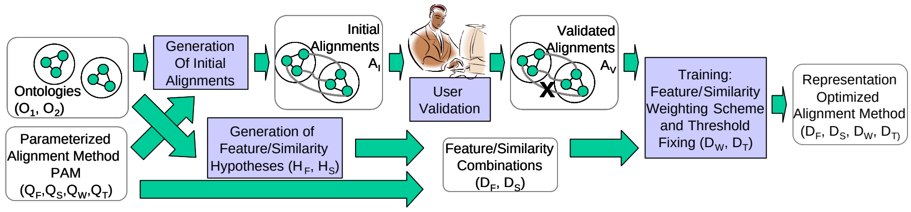

Fig. 1. Detailed Process in APFEL

Core to APFEL is the representation of the generic alignment process. Relevant data structures for representation include: (*i*) Q_F: features engineered (e.g. label, instances, domain), (*ii*) Q_S: similarity assessments corresponding to the features of Q_F (e.g. equality, subsumption), (*iii*) Q_W: weighting scheme for an aggregation of feature-similarity assessments (e.g. weighted averaging), and (*iv*) Q_T: interpretation strategy (e.g. alignments occur if similarity is above the fixed threshold).

Such a declarative representation can be given to a parameterizable alignment method, PAM, for execution. In fact, we can initialize PAM with a representation of different strategies. Thus, an initial alignment function, $\text{align}_{\text{init}}$, may be defined by $\text{align}_{\text{init}}$:=PAM(PROMPT) or $\text{align}_{\text{init}}$:=PAM(QOM). Then, APFEL uses user validations A_V of the initial proposals of $\text{align}_{\text{init}}$.

In general, the described input does not explicitly require a knowledge engineer. The two ontologies, an arbitrary (predefined) alignment method, and the validation of the initial alignments may be processed by a typical user as well.

The output of APFEL is an improved alignment method, $\text{align}_{\text{optim}}$, defined as $\text{align}_{\text{optim}}$:=PAM(APFEL($O_1, O_2, Q_F, Q_S, Q_W, Q_T, A_V$)). The parameters that characterize APFEL($O_1, O_2, Q_F, Q_S, Q_W, Q_T, A_V$) constitute the tuple ($D_F$, D_S, D_W, D_T) with the indices indicating the same as for Q.

3.2 Generation and Validation of Initial Alignments

Machine learning as used in this paper requires training examples. The assistance in their creation is necessary as in a typical ontology alignment setting there are only a small number of really plausible alignments available compared to the large number of candidates, which might be possible a priori.

Therefore, we use an existing parametrization as input to the Parameterizable Alignment Method, e.g. $\text{align}_{\text{init}}$=PAM(QOM) to create the initial alignments A_I for the given ontologies O_1 and O_2. As these results are only preliminary, PAM does not have to use very sophisticated processes: simple features and similarities (e.g. label similarity) combined with an averaging and fixed threshold are sufficient in most cases.

This allows the user to easily validate the initial alignments and thus generate correct training data A_V. If the user further knows additional alignments he can add these alignments to the validated list. Entity pairs not marked by the user are by default treated as disjunct entities. Obviously the quality of the later machine learning step depends on the quality and quantity of the validated alignments done at this point.

3.3 Generation of Feature/Similarity Hypotheses

As mentioned in the introduction it becomes difficult for the human user to decide which features and similarity heuristics make sense in indicating an alignment of two entities. Our approach therefore generates these feature/similarity combinations.

The basis of the feature/similarity combinations is given by an arbitrary alignment method such as PAM(QOM) with which we have achieved good results (see [6]).

Further, from the two given ontologies APFEL extracts additional features H_F by examining the ontologies for overlapping features. At this point domain-specific features are integrated into the alignment process. These features are combined in a combinatorial way with a generic set of predefined similarity assessments including similarity measures for, e.g., equality, string similarity, or set inclusion. Thus, APFEL derives similarity assessments H_S for features H_F. Some feature/similarity combinations will not be very useful, e.g. comparing whether one license number is a substring of another one. However, in the subsequent training step machine learning will be used to pick out those which improve alignment results.

From the feature/similarity combinations of (Q_F, Q_S) and of the extracted hypotheses (H_F, H_S) we derive an extended collection of feature/similarity combinations (D_F, D_S) with $D_F := Q_F \cup H_F$ and $D_S := Q_S \cup H_S$.

3.4 Training

After determining the classification of two entities of being aligned or not (A_V), all validated alignment pairs are processed with the previously automatically generated collection of features and similarities. From each feature/similarity combination a numerical value is returned which is saved together with the entity pair.

Based on these example training alignments A_V we can now learn a classifier which distinguishes between those entities which align and those which are disjunct. Different machine learning techniques for classification (e.g. decision tree learner, neural networks, or support vector machines) assign an optimal internal weighting D_W and threshold D_T scheme for each of the different feature/similarity combinations (D_F, D_S). The machine learning methods like C4.5 (J4.8 in Weka) capture relevance values for feature/similarity combinations. If the latter do not have any (or only marginal) relevance for the alignment, they are given a weight of zero.

From this we finally receive the most important feature/similarity combinations (features D_F and similarity D_S) and the weighting D_W and threshold D_T thereof. With this we can set up the final ontology alignment method which we call align$_{\text{optim}}$:=PAM(APFEL($O_1, O_2, Q_F, Q_S, Q_W, Q_T, A_V$)). Depending on the complexity of the alignment problem it might be necessary to repeat the step of test data generation (based on the improved alignment method) and training.

4 Evaluation

4.1 Evaluation Setting

This paper mainly focuses on an approach to improve methods for the alignment of two ontologies. Neither the learning process APFEL itself nor the quality of the alignment method PAM can be evaluated directly. Therefore, we evaluate the results returned by the learned process, i.e. the quality of the alignments. They are compared to an approach using labels only for comparison and the manually defined alignment process QOM.

We use standard information retrieval metrics to assess the different approaches (cf. [2]): precision $p = \frac{\#correct\_found\_alignments}{\#found\_alignments}$, recall $r = \frac{\#correct\_found\_alignments}{\#existing\_alignments}$, and f-Measure $f_1 = \frac{2pr}{p+r}$. We consider the f-measure as most relevant for our evaluation since it balances precision and recall.

Two different scenarios have been used to evaluate the described machine-learning approach. In the first scenario we use two ontologies describing Russia. Students created these ontologies with the objective to represent the content of two independent travel websites about Russia. The ontologies have approximately 400 entities each. The gold standard of 160 possible alignments was derived by the students who assigned the alignments manually. For the second scenario we have only one ontology, but want to identify equal entities (duplicates) within it. In terms of the problem structure this scenario doesn't differ from a scenario where we want to find equal objects in two ontologies, i.e. an alignment. The ontology describes 2100 bibliographical entities, including 275 manually identified equal entities.

4.2 Results and Lessons Learned

Scenario	Strategy (#/name)	No. of FS	Precision	Recall	F-Measure
Russia	1 Only Labels	1	0.990	0.335	0.422
	2 QOM	25	0.679	0.655	0.667
	3 Decision Tree Learner	7	0.887	0.625	0.733
	4 Neural Net	7	0.863	0.539	0.651
	5 Support Vector Machine	8	0.566	0.636	0.593
Bibliographic	1 Only Labels	1	0.909	0.073	0.135
	2 QOM	25	0.279	0.397	0.328
	3 Decision Tree Learner	7	0.630	0.375	0.470
	4 Neural Net	7	0.542	0.359	0.432
	5 Support Vector Machine	6	0.515	0.289	0.370

Table 1. Results of the Evaluation

From several evaluation runs we have obtained the results in Table 1. Although the precision of an approach based on labels only is very high, the very low recall level leads to a low overall f-measure, which is our key evaluation value. Thus, our key competitor in this evaluation, QOM, receives a lot better results with its semantically rich feature/similarity combinations. To investigate the effectiveness of APFEL, we have tested its different strategies against each other (with 150 training examples for the different learning methods). In both scenarios the decision tree learner returns results better than the two other machine learning approaches, i.e. neural nets and support vector machines. The margin on improvement as compared to QOM in the Russia scenario (6.6 percentage points) and in the Bibliography scenario (7.3 percentage points) is both times very good. Alignments for the Russia scenario are identified precisely. In the bibliographic scenario the alignment method can make extensive use of the learned domain-specific features. Finally, the lower number of feature/similarity combinations (maximum of eight for APFEL vs. 25 for QOM) leads even to an increase in efficiency compared to QOM. To sum up, APFEL generates an alignment method which is competitive with the latest existing ontology alignment methods. However, it is important to apply the correct machine learner and a sufficient amount of training data.

5 Concluding Remarks

Many related approaches have already been mentioned throughout the paper. We here cite an approach also using machine learning, GLUE [3]. However, their learning is restricted on concept classifiers for instances based on instance descriptions, i.e. the content of web pages. From the learned classifiers they derive whether concepts in two schemas correspond to each other. Additional relaxation labeling is based solely on manually encoded predefined rules. Nevertheless, from all ontology alignment approaches their work is closest to APFEL. In [1] the same authors introduce the notion of the use of domain specific attributes, thus restricting their work on databases.

To conclude, with the complexity of the alignment task rising it becomes important to use automated solutions to optimize alignment approaches like PAM without losing the advantages of the general human understanding of ontologies. We contributed to this challenge with our approach APFEL.

Acknowledgements Research reported in this paper has been partially financed by the EU in the IST projects SEKT (IST-2003-506826) and SWAP (IST-2001-34103).

References

1. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. iMAP: discovering complex semantic matches between database schemas. In *Proc. of the 2004 ACM SIGMOD International Conference on Management of Data*, pages 383–394, Paris, France, June 2004.
2. H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proc. of the Second International Workshop on Web Databases (German Informatics Society)*, 2002.
3. A. Doan, P. Domingos, and A. Halevy. Learning to match the schemas of data sources: A multistrategy approach. *VLDB Journal*, 50:279–301, 2003.
4. M. Ehrig and S. Staab. QOM - quick ontology mapping. In F. van Harmelen, S. McIlraith, and D. Plexousakis, editors, *Proc. of the Third International Semantic Web Conference (ISWC2004)*, LNCS, pages 683–696, Hiroshima, Japan, 2004. Springer.
5. J. Euzenat and P. Valtchev. Similarity-based ontology alignment in owl-lite. In *Proc. of ECAI 2004*, pages 333–337, Valencia, Spain, August 2004.
6. T. Hughes. Information interpretation and integration conference, August 2004. http://www.atl.external.lmco.com/projects/ontology/i3con.html.
7. M. Klein. Combining and relating ontologies: an analysis of problems and solutions. In A. Gomez-Perez, M. Gruninger, H. Stuckenschmidt, and M. Uschold, editors, *Workshop on Ontologies and Information Sharing, IJCAI01*, Seattle, USA, August 4-5 2001.
8. N. F. Noy and M. A. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.