Bootstrapping Ontology Alignment Methods with APFEL

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

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

Abstract. Ontology alignment is a prerequisite in order to allow for interoperation between different ontologies and many alignment strategies have been proposed to facilitate the alignment task by (semi-)automatic means. Due to the complexity of the alignment task, manually defined methods for (semi-)automatic alignment rarely constitute an optimal configuration of substrategies from which they have been built. In fact, scrutinizing current ontology alignment methods, one may recognize that most are not optimized for given ontologies. Some few include machine learning for automating the task, but their optimization by machine learning means is mostly restricted to the extensional definition of ontology concepts. With APFEL (Alignment Process Feature Estimation and Learning) we present a machine learning approach that explores the user validation of initial alignments for optimizing alignment methods. The methods are based on extensional and intensional ontology definitions. Core to APFEL is the idea of a generic alignment process, the steps of which may be represented explicitly. APFEL then generates new hypotheses for what might be useful features and similarity assessments and weights them by machine learning approaches. APFEL compares favorably in our experiments to competing approaches.

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. [10, 12, 19]). 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. [1, 7]).

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. Furthermore, it is rather difficult to combine the multiple different substrategies to behave optimally. This is especially the case with increasing complexity of ontology languages or increasing

amounts of domain specific conventions, which should also be included for optimal performance. The second paradigm is often hurt by the lack of instances or instance descriptions, because not in every case an ontology has many instances and in many cases instances exist only for some part of the ontology. Knowledge encoded in the intensional descriptions of concepts and relations is only marginally exploited this way.

Hence, there remains the need to automatically combine multiple diverse and complementary alignment strategies of *all* indicators, i.e. extensional (like similar instances) *and* intensional (like the same position in a taxonomy) 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". The learnt PAM may be applied to ontologies of specific domains, but also for on-the-fly alignment of arbitrary ontologies.

Bootstrapping with APFEL

APFEL is based on four major considerations. First, at the level of *executing the alignment method*, APFEL is based on the general observation that alignment methods like QOM [10] or PROMPT [19] may be mapped onto a generic alignment process (cf. Section 3). Major steps of this generic process include:

1. Feature Engineering, i.e. select small (also domain-specific) excerpts of the overall ontology definition to describe a specific entity (e.g., the `label 'Daimler'` 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., $\text{simil}_{\text{label}}$(`o1:Daimler`,`o2:Mercedes`)=0).
4. Similarity Aggregation, i.e. aggregate multiple similarity assessments 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, some threshold and some interpretation strategy to propose the equality for the selected entity pairs (align (`o1:Daimler`)='⊥').
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`).

Second, at the meta level of *representing an alignment method*, APFEL parameterizes each of these steps by maintaining a declarative representation of features engineered Q_F, similarity assessments Q_S for the features, a weighting scheme Q_W for the aggregation of such similarity assessments and a threshold Q_T to feed into the interpre-

tation strategy (see Section 4.1).[1] In principle APFEL can be applied to every approach based on the presented generic process.

Third, such a declarative representation, e.g. of QOM or PROMPT, can be given to a *parameterizable alignment method*, PAM. In fact, we initialize PAM with the representation of a QOM-like strategy, PAM(QOM), before some initial alignments of two given ontologies are generated through it. The alignments are then handed over to the user for validation (cf. Section 4.2).

Fourth, APFEL generates hypotheses of useful features H_F for a domain-specific pair of ontologies and proposes similarity assessments H_S for these hypotheses (cf. Section 4.3). APFEL uses the validated initial alignments for machine learning the weighting scheme. The aggregation scheme recurs to all feature/similarity combinations under consideration, which are represented by $D_F := Q_F \cup H_F$ and $D_S := Q_S \cup H_S$. Finally, it outputs the weighting scheme D_W and the threshold it has learned D_T (cf. Section 4.4).

The APFEL process is summarized in Figure 2 and will be explained in detail in Section 4. The result of APFEL is a representation of an alignment scheme. The scheme then has been optimized by machine learning to consider the indicators initially used for bootstrapping as well as the newly generated domain/ontology-specific indicators. Thus, it may integrate indicators working at the level of intensional *and* extensional ontology descriptions to result in a comprehensive improved alignment method (cf. Section 4.5).

The paper is structured as follows. In the next section we will explain the foundations for our approach, ontologies and alignments. In Section 3 we describe all steps of the general alignment process in detail. Section 4 illustrates our APFEL approach. In the subsequent Section 5, we evaluate APFEL against various alignment methods, in particular QOM. Before we conclude, we contrast APFEL with further approaches.

2   Foundations

2.1   Ontology

The following short definition describes an ontology structure as used here. In the understanding of this paper an ontology consists of both schema and instantiating data.

An ontology O is defined through the following tuple:

$$O := (C, H_C, R_C, H_R, I, R_I, \iota_C, \iota_R, A)$$

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. Instances and relational instances are connected to concepts resp. relations by the instantiations ι_C resp. ι_R. Additionally one can define axioms A which can be used to infer knowledge from already existing knowledge. An extended definition can be found in [20]. Common languages to represent ontologies are RDF(S) or OWL, though one should note that each language offers different modeling primitives.

---

[1] Unlike done in QOM [10], we do not vary the search step selection, as QOM was about the trade-off between efficiency and effectiveness and in this paper we focus on effectiveness alone. Further, we do not vary iteration strategies to limit the exploration space.

The following fragment of an automobile ontology

$O := (\{automobile, luxury, \ldots\}, \{\ldots\}, \{speed(automobile, INTEGER), \ldots\},$
$\{\ldots\}, \{\ldots\}, \{\ldots\}, \{\ldots\}, \{\ldots\}, \{\ldots\})$

can be represented in OWL as shown in Example 1.

```
<owl:Class rdf:about=''auto:automobile''/>
<owl:Class rdf:about=''auto:luxury''/>
<owl:DatatypeProperty rdf:about=''auto:speed''>
 <rdfs:domain rdf:resource =''auto:automobile''/>
 <rdfs:range rdf:resource=''xsd#INTEGER''/>
</owl:DatatypeProperty>
```

Example 1. Domain Ontology

2.2   Alignment

We here define our use of the term "alignment" similarly to [15]: 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.

Definition 1. *We define an ontology alignment function, align, based on the vocabulary, $\mathcal{E}$, of all terms $e \in \mathcal{E}$ and the set of possible ontologies, $\mathcal{O}$, as a partial function:*
  $align : \mathcal{E} \times \mathcal{O} \times \mathcal{O} \rightharpoonup \mathcal{E}$,
  with $\forall e \in \mathcal{E}_{O_1} (\exists f \in \mathcal{E}_{O_2} : align(e, O_1, O_2) = f \vee align(e, O_1, O_2) = \bot)$.
We write $\mathcal{E}_{O_1}$ if all $e \in \mathcal{E}$ are from ontology O_1. Any entity can either be aligned to exactly one other entity or none.

Apart from one-to-one alignment as investigated in this paper one entity often has to be aligned to a complex composite such as a concatenation of terms (first and last name) or an entity with restrictions (a sports-car is a car going faster than 250 km/h). We refer to [4, 6] for first thoughts on complex alignments. Alignment of axioms has to the best of our knowledge not been a topic of research yet.

3   General Alignment Process

We briefly introduce our definition of the generic alignment process that subsumes all the alignment approaches we are aware of (e.g. PROMPT [19], GLUE [7], QOM
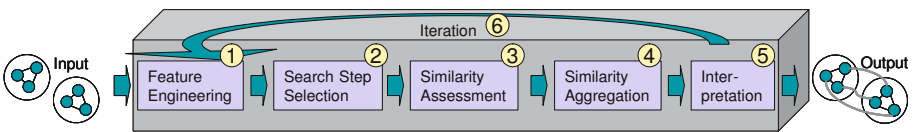

Fig. 1. General Alignment Process in PAM

[10, 11]). This subsumption makes our work a meta-framework valid for many ontology alignment approaches. In this section, we only focus on the definition to the extent that is necessary to understand how APFEL operates on the steps of the generic process. Figure 1 illustrates the six main steps of the generic alignment process. As input, two ontologies are given which are to be aligned. The steps are illustrated through examples where necessary.

1. **Feature engineering** selects only parts of an ontology definition in order to describe a specific entity. Implicitly, [12] made a similar observation. For instance, alignment of entities may be based only on a subset of all RDFS primitives in the ontology. A feature may be as simple as the label of an entity, or it may include intensional structural descriptions such as super- or sub-concepts for concepts (a `sports car` being a subconcept of `car`), or domain and range for relations. Instance features may be instantiated attributes. Further, we use extensional descriptions.

```
<rdf:Description rdf:about=``o1:Daimler''>
    <rdf:type rdf:resource=``auto:automobile''>
    <rdf:type rdf:resource=``auto:luxury''>
    <auto:speed rdf:resource=``auto:fast''>
</rdf:Description>
```

Example 2. Fragment of the First Example Ontology

```
<rdf:Description rdf:about=``o2:Mercedes''>
    <rdf:type rdf:resource=``auto:automobile''>
    <auto:speed rdf:resource=``auto:fast''>
</rdf:Description>
```

Example 3. Fragment of the Second Example Ontology

In our Examples 2 and 3 we have fragments of two different ontologies, one describing the instance `Daimler` and one describing `Mercedes`. Both `o1:Daimler` and `o2:Mercedes` have a generic ontology feature called `type`. The values of this feature are *(i)*, `automobile` and `luxury`, and, *(ii)*, `automobile`, respectively.

Often ontology alignment has to be performed in a specific application of one domain. For these scenarios domain-specific features provide excess value for the alignment process. Returning to our example, the relation `speed` is not a general ontology feature, but a feature which is defined in the automobile domain, e.g. in a domain ontology. Thus it will be important for correctly and only aligning `o1:Daimler` and `o2:Mercedes`.

2. **Selection of Next Search Steps.** The derivation of ontology alignments takes place in a search space of candidate pairs. This step may choose to compute the similarity of a restricted subset of candidate concepts pairs $\{(e, f)|e \in \mathcal{E}_{O_1}, f \in \mathcal{E}_{O_2}\}$ and to ignore others. For the running example we simply select every possible entity pair as an alignment candidate. In our example this means we will continue the comparison of `o1:Daimler` and `o2:Mercedes`.

3. Similarity Assessment determines similarity values of candidate pairs. We need heuristic ways for comparing objects i.e. similarity functions such as on strings [16], object sets [3], checks for inclusion or inequality, rather than exact logical identity. The result lies within a range between 0 and 1. In our example we use a similarity function based on the instantiated results, i.e. we check whether the two concept sets, parent concepts of `o1:Daimler` (`automobile` and `luxury`) and parent concepts of `o2:Mercedes` (only `automobile`), are the same. In the given case this is true to a certain degree, effectively returning a similarity value of 0.5. The corresponding feature/similarity assessment (FS2) is represented in Table 1 together with a second feature/similarity assessment (FS1) based on the similarity of labels. For APFEL we refer to them as Q_F/Q_S assessments. According to the classification by [8] the feature/similarity combinations may be referred to as rule-based alignment approaches.

Table 1. Informal and Formal Feature/Similarity Assessment

FS1: if labels are the same, the entities are also the same to a certain degree
FS2: if parent concepts are the same, the instances are also the same to a certain degree

Comparing	No.	Feature Q_F	Similarity Q_S
Entities	FS1	(label,X_1)	string similarity(X_1, X_2)
Instances	FS2	(parent,X_1)	set equality(X_1, X_2)

4. Similarity Aggregation.   In general, there may be several similarity values for a candidate pair of entities (e, f) from two ontologies O_1, O_2, e.g. one for the similarity of their labels and one for the similarity of their relationship to other terms. These different similarity values for one candidate pair must be aggregated into a single aggregated similarity value. This may be achieved through a simple averaging step, but also through complex aggregation functions using weighting schemes Q_W. For the example we only have to result of the parent concept comparison which leads to: simil(`o1:Daimler`,`o2:Mercedes`)=0.5.

5. Interpretation uses the aggregated similarity values to align entities from O_1 and O_2. Some mechanisms here are e.g. to use thresholds Q_T for similarity [19], to perform relaxation labelling [7], or to combine structural and similarity criteria. simil(`o1:Daimler`,`o2:Mercedes`)=0.5$\geq$0.5 leads to align(`o1:Daimler`)= `o2:Mercedes`. Semi-automatic approaches may present the entities and the alignment confidence to the user and let the user decide.

6. Iteration.   Several algorithms perform an iteration (see also similarity flooding [17]) over the whole process in order to bootstrap the amount of structural knowledge. Iteration may stop when no new alignments are proposed, or if a predefined number of iterations has been reached. Note that in a subsequent iteration one or several of steps 1 through 5 may be skipped, because all features might already be available in the appropriate format or because some similarity computation might only be required in the first round. We use the intermediate results of step 5 and feed them again into the process and stop after a predefined number of iterations.
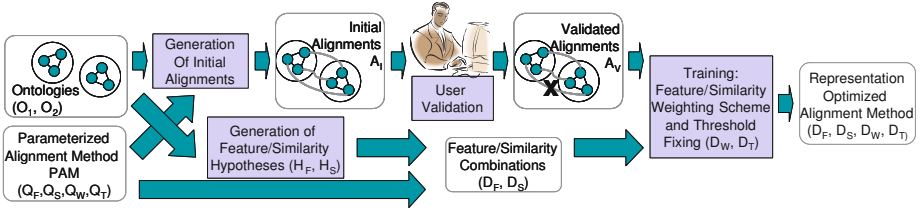
Fig. 2. Detailed Process in APFEL

4   APFEL

In this section it is explained how APFEL works to optimize a given parameterizable alignment method (cf. Figure 2). Data structures are illustrated through white boxes and process steps through colored boxes. We will describe first the data structures, then the process steps. Finally, we show how the PAM resulting from APFEL is applied.

4.1   Data Structures

We here describe the data structures on which APFEL operates. 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 or domain which requires an optimized alignment method.

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, align$_{init}$, may be defined by align$_{init}$:=PAM(PROMPT) or align$_{init}$:=PAM(QOM).

Then, APFEL uses user validations A_V of the initial proposals of align$_{init}$. In general, the described input does not explicitly require an ontology engineer. The two ontologies, an arbitrary (predefined) alignment method, and the validation of the initial alignments may be processed by a typical (domain) user as well, as long as she understands the meaning of the aligned entities.

The output of APFEL is an improved alignment method, align$_{optim}$, defined as align$_{optim}$:=PAM(APFEL($O_1, O_2, Q_F, Q_S, Q_W, Q_T, A_V$)). Parameters characterizing APFEL($O_1, O_2, Q_F, Q_S, Q_W, Q_T, A_V$) constitute the tuple ($D_F, D_S, D_W, D_T$).

Through the optimization step alignment results may change: the result of align$_{init}$(o1:Daimler, O_1, O_2) might be '$\perp$' and the result of align$_{optim}$(o1:Daimler, O_1, O_2) might be o2:Mercedes.

Table 2. Initial Alignments Returned for Validation

Entity 1	Entity 2	Confidence	User Grade
car	car	0.95	to be rated
auto	automobile	0.8	to be rated
wheel	tire	0.6	to be rated
speed	hasSpeed	0.6	to be rated
driver	gear	0.2	to be rated

4.2   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. Presenting every candidate for validation makes the process tiring and inefficient for the human user. Therefore, we use an existing parametrization as input to the Parameterizable Alignment Method, e.g. $align_{init}$=PAM(QOM) to create the initial alignments A_I for the given ontologies. As these results are only preliminary, PAM does not have to use very sophisticated processes: very basic features and similarities (e.g. label similarity) combined with a naïve simple averaging and fixed threshold are sufficient in most cases. Resulting proposed pairs are stored starting with the highest probability alignments as shown in Table 2.

This allows the domain user to easily validate the initial alignments and thus generate correct training data A_V. She does not need to understand the complex ontology concepts i.e. does not need to be an ontology engineer, but has to understand the meanings of the aligned entities. If the user further knows additional alignments she can add these alignments to the validated list. Obviously the quality of the later machine learning step depends on the quality and quantity of the validated alignments at this point.

4.3   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 automatically.

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.

Further, from the two given ontologies APFEL extracts additional features H_F by examining the ontologies for overlapping features. "Overlapping" means that they occur in both ontologies. Currently this implies the same identifier, but very similar features can also be used. These might be additional features from the ontology model such as OWL primitives or special XML datatypes. But at this point also domain-specific features are integrated into the alignment process such as `auto:licensenumber` from an upper-level automobile ontology. The features are then combined in a combinatorial way with a generic set of predefined similarity assessments including similarity

$$\left\{\begin{matrix} extras \\ licensenumber \end{matrix}\right\} \times \left\{\begin{matrix} equality \\ inclusion \end{matrix}\right\} \Rightarrow$$

Comparing	No.	Feature H_F	Similarity H_S
Cars	FS1	(extras,X_1)	set equality(X_1, X_2)
Cars	FS2	(extras,X_1)	subset(X_1, X_2)
Cars	FS3	(license no.,X_1)	equality(X_1, X_2)
Cars	FS4	(license no.,X_1)	substring(X_1, X_2)

Fig. 3. Generation of Additional Hypotheses

measures for, e.g., equality, string similarity, or set inclusion. Thus, APFEL derives similarity assessments H_S for features H_F.

Figure 3 illustrates this process for generating hypotheses for feature/similarity combinations. In the given example two domain attributes `extras` and `license number` are compared using the `equality` and the `inclusion` similarity. All feature/similarity combinations are added for now. Some feature/similarity combinations will not be useful, e.g. FS4, checking whether one license number is a substring of another. However, in the subsequent training step machine learning will be used to pick out those which actually 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 $D_S := Q_S \cup H_S$.

4.4   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 FS set a numerical value is returned which is saved together with the entity pair as shown in Table 3. Further the user validation is added to the table.

We can now apply machine learning algorithms to the automatically generated features D_F and similarities D_S using the example training alignments A_V. More specifically, the numerical values of all feature/similarity combinations are the input for the algorithm. The classification of being aligned or not represents the output. Different machine learning techniques for classification (e.g. decision tree learner, neural networks,

Table 3. Training Data for Machine Learning (including user validation and value returned by each feature/similarity combination FS$_i$)

Entity1	Entity2	FS1	FS2	FS3	FS4	User Grade
car	car	1.0	1.0	0.8	0.0	1
auto	automobile	0.7	1.0	0.7	0.0	1
wheel	tire	0.0	1.0	0.8	0.0	0
speed	hasSpeed	0.7	0.0	0.0	1.0	1
driver	gear	0.2	0.0	0.0	0.0	0

or support vector machines) assign an optimal internal weighting D_W and threshold D_T scheme. However, the number of training alignments and feature/similarity combinations need to correlate to return meaningful results. Machine learning methods like C4.5 further capture relevance values for feature/similarity combinations. If they do not have any (or only marginal) relevance they are given a weight of zero and can thus be omitted. In a decision tree they simply are not present.

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, especially if the initial method was very simple.

4.5   Application in Alignment Process

The final system is parameterized with D_F, D_S, D_W, and D_T. It allows for fully or semi-automatic alignment of two ontologies — and further uses domain-specific optimization of the alignment system. If training data represented general ontologies, the system can be applied to any pair of ontologies for aligning, not only the domain of training ontologies. Depending on the weighting and threshold scheme this may also include an explanation facility which provides evidence why two entities are aligned.

5   Evaluation

5.1   Implementation

The presented approach has been implemented as part of the FOAM framework of ontology alignment and mapping[2]. It is based on Java using the capabilities of the KAON2-framework [14], which can handle OWL-DL ontologies.

5.2   Evaluation Approach

This paper mainly focuses on an approach to create a method for the alignment of two ontologies. The quality of neither the learning process APFEL itself nor the alignment method PAM can be evaluated directly. Therefore, we evaluate the quality of alignments returned by the learned process. They are compared to the manually created alignment process QOM, which has shown very good results in previous experiments [10]. Additionally we evaluated the effect of different numbers of training examples.

5.3   Measures

We use standard information retrieval metrics to assess the approaches (cf. [5]):

---

[2] http://www.aifb.uni-karlsruhe.de/WBS/meh/foam

$$\text{Precision} \quad p \ = \frac{\#correct\_found\_alignments}{\#found\_alignments}$$
$$\text{Recall} \qquad r \ = \frac{\#correct\_found\_alignments}{\#existing\_alignments}$$
$$\text{F-Measure} \ f_1 = \frac{2pr}{p+r}$$

We consider the f-measure as most relevant for our evaluation since it balances well precision and recall. If the focus were laid more onto precision or recall, as may be necessary for specific use cases, slight changes would be necessary in the parameters of the learning step, but this does not jeopardize the general APFEL process.

5.4   Training and Test Data Sets

We here present two of the different scenarios which have been used to evaluate the machine-learning approach.

The first scenario represents the case where we want to align two ontologies based on general ontology features. We want to prove that a good algorithm for aligning very different ontologies can be learned. We rely on eight different ontology pairs and their respective correct alignments as training data. The data has been provided for the alignment contest I3Con[3]. Students created two test ontologies with the objective to represent the content of two independent travel websites about Russia for evaluation. The ontologies have approximately 400 entities each, including concepts (`region`, `river`,...), relations (`has_capital`, `has_mouth`,...), and instances (`Moscow`, `Black_Sea`,...). The gold standard of 160 possible alignments was assigned by the students manually.

In the second scenario we want to optimize the ontology alignment process for one specific domain. This usage scenario is directly taken from the Bibster application, a peer-to-peer system to exchange bibliographic metadata [13]. Thus, we do not use general training data as in the previous scenario, but data from the same ontology domain. For the evaluation the used training alignments are excluded. 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. In this scenario, the two ontologies describe bibliographical entities, such as `articles`, `books`, `theses`, etc. and their respective `authors`, `editors`, or involved `organizations`. For the 2100 entities, 275 duplicates have been manually identified by a domain expert.

One should be aware that the *correct* alignments are also always subjective to a certain degree. Humans normally do not agree on alignments either, often only to 60%, thus making an evaluation result of 100% an unrealistic goal. Further, it is not possible to compare the absolute evaluation results of the two data sets with each other, as the sets differ considerably. For evaluation only the different strategies' results within one set are expressive and may be interpreted.

5.5   Evaluation Strategies

We pursue seven strategies for evaluating the two scenarios.

---

[3] http://www.atl.external.lmco.com/projects/ontology/i3con.html

Table 4. Results of the Evaluation

Scenario	Strategy (#/name)		No. of FS	Precision	Recall	F-Measure
Russia	1 Only Labels		1	0.990	0.335	0.501
	2 QOM		25	0.618	0.596	0.607
	3a Decision Tree Learner	20	1	0.826	0.475	0.603
	3b	50	1	0.819	0.471	0.598
	3c	150	7	0.723	0.591	0.650
	4 Neural Net	150	7	0.777	0.485	0.597
	5 Support Vector Machine	150	8	0.509	0.572	0.539
Bibliographic	1 Only Labels		1	0.909	0.073	0.135
	2 QOM		25	0.279	0.397	0.328
	3a Decision Tree Learner	20	1	0.047	0.280	0.080
	3b	50	2	0.456	0.246	0.318
	3c	150	7	0.630	0.375	0.470
	4 Neural Net	150	7	0.542	0.359	0.432
	5 Support Vector Machine	150	6	0.515	0.289	0.370

- The first strategy simply aligns based on the equality of labels. This is a strategy used for example in the original PROMPT tool [19].
- The second strategy applies a variety of general ontology alignment feature/similarity combinations and an aggregation thereof (QOM). They have been exclusively created by an ontology engineer understanding the domain of knowledge modeling with ontologies. Further, the combinations were assigned manual weights and an optimized threshold (see [10]).
- The remaining strategies represent the APFEL approach. The third strategy uses a C4.5 (J4.8 in Weka) decision tree learner. We took a varying number of 20, 50, and 150 training examples from the correct alignments to further investigate the effect of different quantities of training examples. Half of the examples were positives and half were negatives. For all machine learning approaches we use the well-known WEKA machine learning environment[4].
- The next strategy uses a neural net based on 150 examples.
- And the last strategy was to train a support vector machine, with 150 examples.

5.6   Results and Lessons Learned

From several evaluation runs we have obtained the results in Table 4. 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 f-measure with its semantically rich feature/similarity combinations.

   To investigate the effectiveness of APFEL, we have first tested the 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 decision tree learner delivers the best f-measure. The margin on improvement as compared to

---

[4] http://www.cs.waikato.ac.nz/ ml/weka/

QOM in the Russia scenario (4.3 percentage points) and in the Bibliography scenario (14.2 percentage points) is both times very good. Alignments for the Russia scenario are identified precisely. Similarly as in the manual approach `labels` were given a very high rate, but surprisingly `domain` and `range` differentiate concepts better than the obvious `sub-classes`. In the bibliographic scenario the alignment method can make extensive use of the learned domain-specific features e.g. it identifies the attribute `last_name` as being highly relevant to find identical authors and rates it higher than e.g. the `middle_initial`. Finally, the lower number of feature/similarity combinations (maximum of eight for machine learning vs. 25 for QOM) leads even to an increase in efficiency compared to QOM.

Second, we have considered the learning rate (see 3a-3c in Table 4). Quality increases with the number of training examples rising, somewhat leveling off at a good value. Unfortunately due to the complex structures of ontologies with many possible feature/heuristics combinations, a high absolute number of training examples is required to fully capture their semantic value for alignment. In the research domain of ontology alignment with its current lack of real big examples this is a challenge. However, once learned it can be transferred to ontology alignment problems in the same domain/ontology model without further learning effort.

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.

6   Related Work

In [8] schema matching approaches for the database community are split into rule-based and learning-based techniques. In this paper we have shown how to apply learning techniques on-top of a rule-based approach. To contrast our approach we use their classification in the following.

The tools PROMPT and AnchorPROMPT [19] use the similarity of labels and to a certain extent the structure of ontologies, creating alignment rules thereof. The concrete algorithm is set through the tool developers manually. Adaptations to new ontological constructs or even domain-specific features can not be incorporated. In their tool ONION [18] the authors use rules and inferencing to execute alignments, but the inferencing is again based on initially manually assigned alignments or simple similarities. An interesting field of future research are complex alignments, which we do not consider yet in this paper. These cover alignments e.g. based on the concatenation of two fields such as "first name" and "last name" to "name" (cf. COMA[6]). [2] finally present an approach for semantic alignment based on SAT-solvers. In their approach an alignment can only be created if there are no inherent semantic rules restricting this, thus making it an approach based on exact semantics rather than on heuristics as in our work. Nevertheless in all these works one faces the difficulty to predict which strategy of aligning entities is most successful for a given pair of ontologies. The optimization strategy APFEL pointed out in this paper could enhance these existing approaches.

[7] use machine learning in their approach GLUE. From all ontology alignment approaches their work is closest to APFEL. However, their learning component is restricted

on concept classifiers for instances based on instance descriptions, i.e. the textual content of web pages, or their naming. From these two learned concept classifiers they derive whether concepts in two schemas correspond to each other, whereas our approach focuses on learning parameters for a general alignment process. The GLUE machine learning approach suits a scenario with extensive textual instance descriptions, but may not suit a scenario focused more onto ontology structures. Further, relations or instances can not be directly aligned with GLUE. The additional relaxation labeling, which takes the ontological structures into account, is again based solely on manually encoded predefined rules. Finally, in [9] the same authors introduce the notion of the use of domain specific attributes, thus restricting their work on databases. However, the inclusion of domain typical structures has not been topic of their work while it is provided by APFEL.

7   Conclusion

High-quality semantic alignment between ontologies is a necessary precondition to establish interoperability between agents or services using different ontologies. Recent work suffers from the problem that it is impossible to predict which strategy of aligning entities is most successful, given an often semantically and structurally rich domain ontology.

Thus, we have developed a method called APFEL ("Alignment Process Feature Engineering and Learning") that applies machine learning for creating an alignment method that produces a better quality than an initial alignment strategy it starts with.

The involvement of users happens in two phases. Initially, users provide domain ontologies and a simple general alignment method for getting started. During the process, users need to evaluate the generated initial alignments. However, there is no requirement for ontology engineers being involved. I.e., users without specific knowledge about ontology engineering are able to use our approach.

APFEL iteratively bootstraps a new alignment method which is optimized for the input ontologies. This process has been presented in detail on the preceding pages. The resulting alignment method can then be used to automatically align ontologies, depending on the training set-up either general arbitrary ontologies or domain-specific ones. From the evaluation results we have obtained, we see that our initial hypothesis of using machine learning to gain a better alignment approach was fulfilled. The machine learned process outperforms the various manual approaches.

References

1. R. Agrawal and R. Srikant. On integrating catalogs. In *Proceedings of the Tenth International Conference on the World Wide Web (WWW-10)*, pages 603–612. ACM Press, 2001.
2. P. Bouquet, B. Magnini, L. Serafini, and S. Zanobini. A SAT-based algorithm for context matching. In *Proc. of the Fourth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'2003)*, Stanford University (CA, USA), June 2003. Springer.

3. T. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994.

4. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. iMAP: discovering complex semantic matches between database schemas. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pages 383–394, Paris, France, June 2004.

5. H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proceedings of the Second International Workshop on Web Databases (German Informatics Society)*, 2002.

6. H.-H. Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In *Proceedings of the 28th VLDB Conference*, Hong Kong, China, 2002.

7. A. Doan, P. Domingos, and A. Halevy. Learning to match the schemas of data sources: A multistrategy approach. *VLDB Journal*, 50:279–301, 2003.

8. A. Doan and A. Y. Halevy. Semantic-integration research in the database community. *AI Magazine*, pages 83–94, March 2005.

9. A. Doan, Y., Lu, Y. Lee, and J. Han. Object matching for data integration: A profile-based approach. In *Proceedings of the IJCAI-03 Workshop on Information Integration on the Web*, Acapulco, Mexico, August 2003.

10. M. Ehrig and S. Staab. QOM - quick ontology mapping. In F. van Harmelen, S. McIlraith, and D. Plexousakis, editors, *Proceedings of the Third International Semantic Web Conference (ISWC2004)*, LNCS, pages 683–696, Hiroshima, Japan, 2004. Springer.

11. M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In *Proceedings of the First European Semantic Web Symposium, ESWS 2004*, volume 3053 of *Lecture Notes in Computer Science*, pages 76–91, Heraklion, Greece, May 2004. Springer Verlag.

12. J. Euzenat and P. Valtchev. Similarity-based ontology alignment in owl-lite. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI2004)*, pages 333–337, Valencia, Spain, August 2004.

13. P. Haase et al. Bibster - a semantics-based bibliographic peer-to-peer system. In F. van Harmelen, S. McIlraith, and D. Plexousakis, editors, *Proceedings of the Third International Semantic Web Conference (ISWC2004)*, LNCS, pages 122–136, Hiroshima, Japan, 2004. Springer.

14. U. Hustadt, B. Motik, and U. Sattler. Reducing SHIQ-description logic to disjunctive datalog programs. In *Proceedings of Ninth International Conference on Knowledge Representation and Reasoning 2004*, pages 152–162, Whistler, Canada, June 2004.

15. 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, 2001.

16. I. V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 1966.

17. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of the 18th International Conference on Data Engineering (ICDE'02)*, page 117. IEEE Computer Society, 2002.

18. P. Mitra, G. Wiederhold, and M. Kersten. A graph-oriented model for articulation of ontology interdependencies. In *Proceedings of the Conference on Extending Database Technology 2000 (EDBT'2000)*, volume 1777, pages 86+, Konstanz, Germany, 2000.

19. 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.

20. G. Stumme et al. The Karlsruhe view on ontologies. Technical report, University of Karlsruhe, Institute AIFB, 2003.