Rough Description Logics for Modeling Uncertainty in Instance Unification

Michel C.A. Klein[1], Peter Mika[2], and Stefan Schlobach[1]

[1] Vrije Universiteit Amsterdam, {michel.klein|schlobac}@few.vu.nl
[2] Yahoo! Research Barcelona, pmika@yahoo-inc.com

Abstract. Instance-unification is a prime example for uncertainty on the Semantic Web, as it is not always possible to automatically determine with absolute certainty whether two references denote the same object or not. In this paper, we present openacademia, a semantics-based system for the management of distributed bibliographic information collected from the Web, in which the Instance Unification problem is ubiquitous. Our tentative solution is Rough DL, a simple extension of classical Description Logics, which allows for approximations of vague concept. This shows that already a simple formalism for dealing with uncertain information in a qualitative way can provide an elegant solution to practical problems on the Semantic Web.

1 Introduction

When information is gathered from the Web it often occurs that multiple descriptions of the same resource are found. In that case the duplicate resources should be identified and their descriptions have to be combined.

Failing to effectively deal with duplicate results negatively affects the workings of all search engines. In a search system for publications instance unification is important for at least two types of information: persons and publications. If coreferences of persons [3, 4] are not resolved one will obtain an incomplete list of publications when querying for publications of a specific person (e.g. because the system does not recognize that the authors 'John Smith' and 'John J.B. Smith' are the same person. One may face the opposite situation of receiving irrelevant results, such as when a system assumes that authors with the same name are the same person, as is common in most existing NLP based publication search engines such as Google Scholar and CiteSeer. On the other hand, if publications are not unified [8, 6] the result will contain duplicates, which makes browsing the results difficult and obscures publication counts, an important statistic in academia. The problem of finding equivalent instances in this case is usually referred to as "coreference resolution" or "instance unification".

The most common way of representing the results of instance unification is to declare the objects to be logically equivalent. This has the consequence that all properties of one resource are also properties of the other resource. However, this implementation has several drawbacks. First, it often represents a **logical**

overcommitment. Only in very limited cases are we absolutely *certain* that two instances are equivalent, in most cases we only have some partial evidence that two descriptions refer to the same object (e.g. name similarity). Further, it is not possible to **distinguish between different levels of confidence** in similarity relations. Moreover, transitivity of equivalence often causes an **undesired propagation of equivalence** over similarity relations.

In this paper, we will introduce an alternative to complete instance unification, which allows for reasoning over gradually weakening notions of similarity. Our tentative solution is an extension to standard DL that can be used for defining approximations of concepts without increasing the complexity of the language.

We illustrate the use of this language in openacademia,[1] an open source web-based system for collecting, aggregating and querying publication metadata in a group or community setting. openacademia offers an interactive, AJAX-based search interface for querying publications by a combination of facets. Query results can be visualized in a number of ways, including the possibility to generate various dynamic HTML representations that can be easily inserted into personal homepages or institutional publication pages. Integrating descriptions of similar persons and publications is an important task of this system, which is in this context sometimes called *smushing* [5].

In the subsequent sections, we introduce the language for defining approximations, and apply it to model different levels of similarity of persons and publications in openacademia. The flexible instance unification using Rough DL illustrates how already rather simple mechanisms for dealing with uncertainty in a qualitative way can be used to elegantly solve practical problems on the Web.

Of course, we do not claim that our practical problem could not have been solved by other, for example more quantitative formalisms. However, we believe that the simplicity of our approach makes it an attractive alternative for dealing with uncertainty on the Semantic Web.

2 Rough DL

In [10] we presented a new paradigm to represent and reason about similarity of instances in a qualitative way called *rough Description Logics* ($\mathcal{RDL}$).[2] This language is an obvious candidate for modeling similarity and reasoning about classes of de-referenced objects. Here, we introduce an adapted version of $\mathcal{RDL}$, which is based on similarity rather than equivalence relations.

Definition 1. *A relation is called a* similarity (or tolerance) relation *if it is reflexive and symmetric. An* equivalence relation *is a transitive similarity relation.*

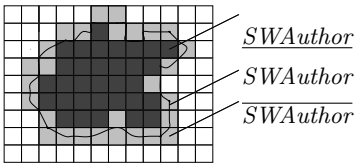
As equivalence relations extend similarity relations, we define Rough DL using the latter. We will use the notation $\overline{C}^{sim}$ and $\underline{C}_{sim}$ to describe the approximations of a class C with respect to a similarity relation sim. We will omit the

[1] `http://www.openacademia.org`

[2] As the relation between OWL and Description Logics is well established, we only introduce rough DL. The extension to rough OWL is conceptually trivial.

lower-script and upper-script sim whenever the choice of relation is irrelevant. The intuition regarding $\overline{C}^{sim}$ is that it denotes the set of all elements that are **possibly** in C, whereas $\underline{C}_{sim}$ is meant to describe all elements **definitively** in C. Often such an operator is useful when C cannot be specified in a crisp way. By way of the approximation(s) we can at least restrict C with an upper, and a lower, bound.

An illustrating example The following picture illustrates the general idea. In the spirit of Rough Set theory [7], two concepts approximate an under-specified, vague, concept as particular sub- and super-concepts. Suppose that we want to define *SWAuthor* as the class of all Semantic Web authors. This is a vague concept.



Each square denotes a set of domain elements, which cannot further be discerned by some available criterion at hand. The encircling line denotes the set of Semantic Web authors, i.e., the vague concept which we are incapable to formally define. If we capture this lack of criteria to discern between two objects as a indiscernibility relation `indis`, we can formalize the upper approximation as the authors that are indiscernible from at least one Semantic Web author.

$$\overline{\texttt{SWAuthor}} \equiv \{aut_1 \mid \exists\ aut_2\texttt{: indis(}aut_1\texttt{,}aut_2\texttt{) \& }aut_2 \in \texttt{SWAuthor}\}.$$

Similarly, we can define the lower approximation as the set of authors containing all, and only those authors, for which it is known that all indiscernible authors must be Semantic Web authors.

$$\underline{\texttt{SWAuthor}} \equiv \{aut_1 \mid \forall\ aut_2\texttt{: indis(}aut_1\texttt{,}aut_2\texttt{) } \rightarrow aut_2 \in \texttt{SWAuthor}\}$$

In our picture, the upper approximation is depicted as the union of the dark squares (the lower approximation), and the gray squares, the boundary. Note that in our example, following the literature on Rough Sets, the similarity of objects is determined by the indiscernibility of resources. This is an equivalence relation, which makes it appropriate to denote the sets of indiscernible instances as disjoint squares.

This intuition suggests two uses for Rough DL: first as a modeling language for representing vague knowledge and, secondly, as a language to query over similarity in a domain.

Modeling vague concepts Even if it is impossible to formally define a concept such as *SWAuthor*, we can often specify the approximations. The class of Semantic Web authors cannot be defined in a crisp way, but it is easy to think of an upper approximation (the possible Semantic Web authors, e.g. all authors having published in a Semantic Web conference or Journal). Rough DL semantics enforce restrictions on the class `SWAuthor` indirectly. We will discuss modeling with Rough DL concepts later in more detail.

Qualitative querying over similarities In the case of instance unification Rough DL can be used for querying classes of objects, and objects that were identified as being similar. Suppose we have a particular author `author` who is uniquely identifiable, say via his FOAF profile. Now, an algorithm Alg for object de-referencing creates a relation sim_{Alg} of pairs $(\texttt{author}_1, \texttt{author}_2)$. Based on this relation each algorithm for referencing induces a set $\texttt{Possibly}_{Alg}(\texttt{author})$ for each author `author`, i.e. a set of objects of the domain $\mathcal{U}$ which possibly correspond to this particular author `author`, with the formal definition:

$$\texttt{Possibly}_{Alg}(\texttt{author}) = \{i \in U \mid \exists j \in U : (i,j) \in sim_{Alg} \ \& \ j \in \texttt{oneOf}(\texttt{author})\},$$

which corresponds almost exactly to the formal semantics of an upper approximation. Most of the remainder of this paper will be about using Rough DL for querying ontologies with explicit similarities.

2.1 Semantics of Rough DL

Using this property we can define the semantics of the approximations formally:

Definition 2. *Let a rough interpretation be a triple $\mathcal{I} = (U, R^{\sim}, \cdot^{\mathcal{I}})$, where U is a universe, $\cdot^{\mathcal{I}}$ an interpretation function, and $R^{\sim}$ an equivalence relation over U. The function $\cdot^{\mathcal{I}}$ maps $\mathcal{RDL}$ concepts to subsets and role names to relations over the domain U. It extends to the new constructs as follows:*

- $(\overline{C})^{\mathcal{I}} = \{i \in U \mid \exists j \in U : (i,j) \in R^{\sim} \ \& \ j \in C^{\mathcal{I}}\}$
- $(\underline{C})^{\mathcal{I}} = \{i \in U \mid \forall j \in U : (i,j) \in R^{\sim} \rightarrow j \in C^{\mathcal{I}}\}$

The semantics of the lower approximation is defined as usual as the dual operator $\underline{\texttt{C}}_{sim} = \neg\overline{\neg C}^{sim}$ with its respective semantics. Depending on the specifics of the similarity relation, these semantics enforce powerful terminological consequences. In [10] we discuss a number of them, here we have to restrict ourselves to two relatively simple examples: Given an ontology $\mathcal{O} = \{\overline{\texttt{SWAuthor}}^{eq} \sqsubseteq \texttt{Author}\}$ where eq is an equivalence relation, it follows that $\mathcal{O} \models \overline{\texttt{SWAuthor}}^{eq} \sqsubseteq \underline{\texttt{Author}}_{eq}$. What does this mean? It means that if any possible Semantic Web author is an author, it must be a typical author. Another example is the non-existence of a definitively non-typical Semantic Web author. Let the non-typical Semantic Web authors be defined as the Semantic Web authors that are not typical Semantic Web authors, i.e. we add $\texttt{NTSWAuthor} \sqsubseteq \texttt{SWAuthor} \sqcap \neg\underline{\texttt{SWAuthor}}_{eq}$ to $\mathcal{O}$. Rough DL semantics implies that there can be no definitely non-typical Semantic Web authors, i.e. that $\mathcal{O} \models \underline{\texttt{NTSWAuthors}}_{eq} = \bot$.

Related to these semantic consequences is the question of reasoning support, i.e. the existence of tools that can calculate consequences such as the ones discussed above in reasonable time. This points to the nice property of Rough DL being a conservative extension of OWL, in the sense that any Rough DL ontology can be translated into a logically equivalent OWL ontology. This means that reasoning for our language comes for free, as we can use standard reasoners to calculate class hierarchies, consistency and all instances of a particular class. The latter is the reasoning most needed in openacademia.

We can now relate these semantics of Rough DL to our example of instance unification given before. If we identify the class `TBL` with the singleton set `t.b-l`, i.e. define `TBL` to be equivalent to `oneOf(t.b-l)`, the set $\texttt{Possibly}_{Alg}\texttt{(t.b-l)}$ semantically corresponds to the upper approximation of `TBL` according to the indiscernibility relation sim_{Alg}.

Adapting Rough DL for openacademia As already mentioned, for application of Rough DL in openacademia we need a slightly different formalism from the one introduced in [10] and described above. First, we use similarity relations in addition to equivalence relations, and secondly, we want to apply different similarity relations and approximations, as well as hierarchies on both.

Similarity versus equivalence In openacademia, approximations based on similarity relations are used in addition to equivalence relations. Some smushing algorithms indeed produce equivalences, e.g. when two instances are identified through equivalence of the value of an inverse functional property. But even in logically weaker cases, there will be methods which indicate most likely equivalence between objects.

On the other hand, there are weaker methods, which will give indications for similarity, and which are not transitive. A simple example is edit distance: a similarity between two instances which is defined by an edit distance smaller or equal to 1 is non-transitive.

Hierarchies on similarities & approximations In an application such as openacademia there is no unique best way to identify co-reference of instances. This means that there will usually be several algorithms, such as the ones described in the following section, which produce several possible similarity relations. Often inclusion properties of such relations are easily created, and are often even more meaningful than quantitative values. In an RDF(S) based framework we can make use of hierarchies on relations to specify confidence in smushing algorithms in a qualitative way.

A simple logical consequence of specifying similarities in hierarchies of properties is that it implies hierarchies of the approximations. More concretely, suppose that two algorithms A_1 and A_2 produce two similarities $\texttt{oa:similarTo}A_1$ and $\texttt{oa:similarTo}A_2$ where, by construction, $\texttt{individual}_1$ $\texttt{oa:similarTo}A_1$ $\texttt{individual}_2$ implies that $\texttt{individual}_1$ $\texttt{oa:similarTo}A_2$ $\texttt{individual}_2$. This is a typical case, as smushing algorithms often include results of other algorithms. In this case, it can be shown that an upper approximation based on $\texttt{oa:similarTo}A_1$ is more specific than an upper approximation based on $\texttt{oa:similarTo}A_2$.

We make use of this property to construct hierarchies of approximations based on the underlying similarity relations, which can be very useful for controlled query relaxation.

3 Using Rough DL in openacademia

The current interface of openacademia allows to query for publications using combinations of different criteria, such as “author”, “title”, “year”, “type” and

"group". With respect to the author criteria, users can provide a string that is matched to a part of the author name.

This is a suboptimal solution when one wants to have precise control over the search results, as there is no way to distinguish between publications of different authors with a similar name. Regardless of instance unification, the result will be a mix of publications of possibly different persons.

A first requirement for controlled instance unification is to search by the URI of the resource instead of its label. For example, the system could search by label and return a list of publications. Then, the user could select a specific instance of an author in the result list, whose URI is used for the subsequent searches.

openacademia currently uses several methods to determine similarity between authors and publications, which are sometimes called *smushing algorithms*.

1. The most certain way to determine the equivalence of two resources is by comparing the values for their `owl:InverseFunctionalProperty`'s. When two resources have the same value for such a property they can be considered as equivalent. The FOAF-specification defines a number of properties, including `foaf:mbox` and `foaf:homepage` as inverse functional.
2. Another method is based on the comparison of the labels of resources. In openacademia we use several heuristics with different certainty to determine possible equivalences. For example, we consider instances of `foaf:Person` as unification candidates if both their first and last names match exactly (i.e. the string is identical), or if their last name and their initials match, or if their last name and first name are within a certain edit-distance.
3. An alternative method exploits the similarity of related resources. If, e.g., two instances of `swrc:Publication` are determined to be equivalent, we assume that the resources in the author-list are also equivalent.

Different from related work that focuses on learning the rules of smushing (e.g. [1]), smushing is an iterative reasoning process in openacademia. The instance matches found in one iteration can be used to discover new matches in subsequent iterations. Iterative reasoning is even a requirement if the smushing rules are co-dependent, such as the case when one would like to infer similarities of persons based on similarities of publications and vice versa, similarities of publications based on similarities of their authors.

To reflect the fact that different algorithms have a different certainty, we add *similarity* statements of the form $\texttt{individual}_1$ `oa:similarTo`X $\texttt{individual}_2$ to the repository. Each individual must be identified by a URI and `oa:similarTo`X is a similarity relation of instances returned by one of the algorithms discussed.

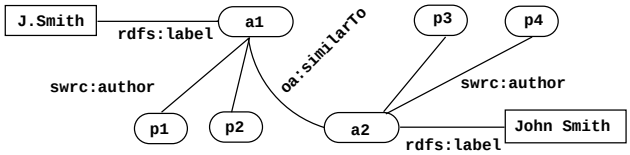
3.1 Benefits of Rough DL

In the context of openacademia, Rough DL has two functions: it offers an appealing conceptual framework for querying over similarities, and it provides the possibility to model vague concepts, such as typical Semantic Web authors. We will discuss both features in more detail.

Querying over similarities Using similarities for object-dereferencing is a relatively obvious proposal as the problems of ad-hoc solutions (such as using `owl:sameAs`) are known. However, most people decide to use the slight over-commitment of `owl:sameAs` in order to be able to continue to use the automatic reasoning support for OWL, i.e. in order to avoid having to deal with similarities explicitly.

Rough DL offers an attractive alternative, as it provides a conceptually elegant framework for querying an RDF(S) repository including similarity relations by two simple operators ($\dot{-}$) and ($\underline{\cdot}$), the approximations. Let us start with a simple example of how Rough DL can help in order to formulate concise queries over graphs including similarities.

Suppose we have identified two resources, `a1` and `a2`, each of type `foaf:Person`. Each resource is connected via a `swrc:author` property to a number of resources of type `swrc:Publication` (`p1`...`p4`). Besides this, `a1` and `a2` each have an `rdfs:label`. One of the similarity heuristics discovered a similarity between the labels of `a1` and `a2`, which is represented by a property `oa:similarTo` between both resources.



The obvious query to take from this graph is to find all publications of the resource `a1`, uniquely identified by the URI `<http://www.uni1.edu/~personA/pubs.bib#john_smith>`, and every resource that is similar.

Formulating this as instance checking in Rough DL is simply to ask for all instances of class $\overline{\texttt{oneOf}\{\ldots\texttt{\#john\_smith}\}}$ (for the resource). Already for such a simple Rough DL query, the corresponding SeRQL query requires explicit knowledge of the structure of the graph, and of the similarity relation used.

```
SELECT distinct Pub FROM
  {Pub}    swrc:author {Person},
  {Person} oa:nameSimilarTo3
       {<http://www.uni1.edu/~personA/pubs.bib#john_smith>}
```

Another example exploits similarities between publications. Suppose that we add different kinds of similarity relations between publications. For example, two publications could be connected via a property `oa:hasJointAuthors` if they share at least two authors. Or, another possibility, they can be connected via `oa:hasRelatedKeywords` if their keywords are related according to some metric, e.g. because the keywords are semantically close to each other in some topic hierarchy. One could even add similarity relations based on the textual overlap of the abstract.

Now, the Rough DL framework allows queries for upper approximations — according to a specific type of similarity — of publications that fulfill specific criteria. For example, if we use the "author overlap" similarity, we could query for

the upper approximation of papers with "OWL" as a keyword by simply asking for the instances of the Rough DL class $\overline{\texttt{restriction(keyword hasValue("OWL"))}}$. When we have `oa:hasJointAuthors` in place as a similarity relation we get as the result all papers of authors that together have published papers with OWL as a keyword. Again, the corresponding SeRQL query is simple, but we could easily use a different similarity measure without requiring any change for the user.

```
SELECT DISTINCT Pub FROM {Pub}
   oa:hasJointAuthors {Other}, WHERE
{Other} IN (SELECT Pub FROM {Pub} swrc:keyword {"OWL"})
```

Modeling vague concepts in OA Up to now we discussed the use of Rough DL for formulating queries over an RDF(S) repository with similarity relations. But of course, the language can also be used to model vague concepts directly in the repository. Imagine that one wants to model Semantic Web conferences and authors, obviously terms that describe vague concepts for which it will be impossible to find commonly agreed upon definitions. What is possible, on the other hand, is to define approximations for both classes. Most people would agree that there are three prototypical Semantic Web conferences: the International and European Semantic Web conferences (ISWC and ESWC), and the World-Wide Web conference (WWW). Defining the lower approximation of a class `SWconference` can then be done simply as $\underline{\texttt{SWconference}} = \underline{\texttt{WWW}} \sqcup \underline{\texttt{ISWC}} \sqcup \underline{\texttt{ESWC}}$ where the conferences are described as classes (e.g. ISWC) containing at least one uniquely identifying resource (e.g., `ns:iswc`). Here, we chose the lower approximations of the resources for the conferences as we want to avoid ambiguity through spelling variants or other forms of synonymy (e.g. ESWC also refers to the Electronic Sports World Cup).

Semantic Web authors can now be defined as authors having published at *possible* Semantic Web conference, i.e.

$$\texttt{SWauthor} = \exists\ \texttt{publishedIn}.\overline{\texttt{SWconference}}$$

Even though Rough DL is a conservative extension of OWL DL this example shows that modeling the same information without approximation operators would be extremely cumbersome. Using Rough DL, and the reasoning machinery that comes for free, thanks to the translation back to OWL, allows queries such as for all possible Semantic Web authors $i : \overline{\texttt{SWauthor}}$?, which even for this simple example is non-trivial on a larger data set.

Furthermore, for the Rough DL fragment built on the OWL DL dialect, the usual reasoning services, such as query entailment, satisfiability checking or subsumption hierarchies can be easily calculated.

For openacademia querying with Rough DL is the more prominent application, and we have not yet pursued modeling of rough concepts in openacademia. Technically, and conceptually, it is easy to add Rough DL axioms to the Sesame repository. By adding rules, part of the OWL semantics can be captured, but completeness cannot be achieved. A more detailed study of this, e.g., considering the alternative semantics proposed in [12], is outside the scope of this paper.

Therefore, although we are also convinced that modeling approximate concepts will significantly improve the ease of use of openacademia, the focus in this paper will be on querying from now on.

3.2 Technical Issues

The application of Rough DL in openacademia amplifies a discrepancy of two Knowledge Representation frameworks that is present in many practical approaches to the Semantic Web. With query languages such as SeRQL or SPARQL for RDF(S) ontologies and the advent of robust and fast RDF repositories, efficient data access is now made possible even for very large data sets. On the other hand, the expressivity of OWL makes it possible to model ontological knowledge in very elegant ways, which is needed for many realistic applications. Unfortunately, theoretically both paradigms are less easily integrated than one would hope for, and than could be expected at first glance.

1. A first issue is the use of a query-language such as SeRQL for a repository containing OWL statements.
2. The second problem to be addressed is the question of Open- versus Closed-World Assumption.

For lack of space both issues can only be discussed briefly.

First, to make use of the best of both worlds, many people include OWL ontologies in RDF repositories, and query those with traditional RDF query languages. The problem with such an approach is that completeness cannot be guaranteed in general. For openacademia we do the same: particular knowledge is represented in OWL (e.g. defining a property as transitive or functional), but SeRQL is used for querying. In the case of openacademia, however, the problem of incompleteness can be circumvented. This is done by including parts of the OWL semantics in the Sesame inference engine (in the style of [12]), and by encoding parts in the queries.

Secondly, querying a triple store such as Sesame usually employs a Close-World assumption, i.e. a universally quantified statement is evaluated as true if all known instances in the relation have the required property. This is different than DL, where an Open-World Assumption is taken. For the lower approximation this means that the DL interpretation differs from the interpretation of a natural SeRQL encoding. As a DL query for lower approximations will only return relevant results when there are explicit universal statements (or approximations) in the ontology, our current research focuses on the use of the upper approximation for querying.

4 Case study

To illustrate the benefits of Rough DL descriptions in practice, we show the effect of applying different approximations in openacademia. For this, we use the approximation interface as shown in Figure 1. This interface translates a restricted set of Rough DL queries into SeRQL.

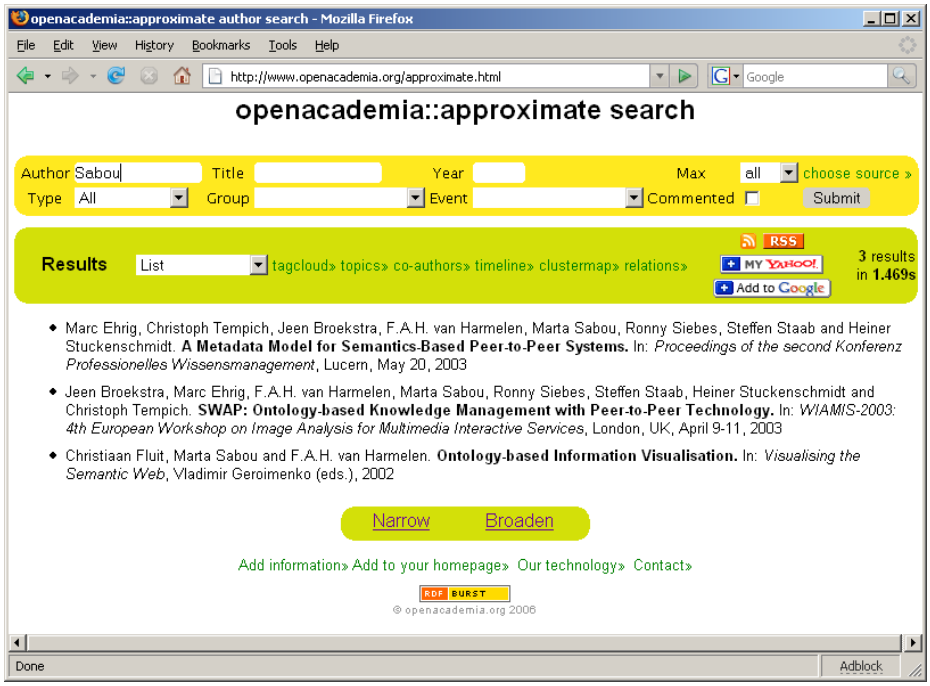


Fig. 1. The approximation interface, allowing for broadening or narrowing the author concept.

We apply Rough DL for author similarity, using a hierarchy of similarities. Suppose we want to query for publications of "Marta Sabou" with several entries in our database. We start with a URI `<http://www.uni1.edu/~personA/pubs.bib#marta_sabou>` in our own BibTeX files, which gives high confidence that the resource represents the right person.

No approximation Without approximation, a query for the publications of this resource results in 3 publications, which were all specified in `<http://www.uni1.edu/~personA/pubs.bib>`.

Exploiting inverse functional properties Smushing adds `oa:nameSimilarTo1` statements between all resources with the same value for an inverse functional property. As Marta's email is listed in her FOAF-profile, an `oa:nameSimilarTo1` statement is added between the original resource and `<http://www.uni1.edu/swhome/person/marta>`, an RDF representation of a personnel database. When querying for the upper approximation of the resource, the search results now include the publications on Marta's homepage.

Exact match of fullname A second level of approximation uses the label of the resources of type `foaf:Person`. When the first and lastname of a person exactly match, a `oa:nameSimilarTo2` is added. This results in similarity statements between the original resource and `<http://www.uni2.edu/~personB/`

biblio/bnaic2002.bib#marta_sabou> and <http://www.uni3.edu/~personC/pubs.bib#marta_sabou>. When using this property as similarity in the Rough DL framework, the search result contains 7 publications.

Exact match of lastname and initial The next level of approximation exploits the oa:nameSimilarTo3 statements that are added when both the lastnames match and the initial of one resource matches the first character of the firstname of another resource. This results in similarity statements to the resource <http://www.uni3.org/~personD/publications.bib#m_sabou> with the label "M. Sabou" and the resource <http://www.uni1.edu/swhome/person/marta> with the label "M.R. Sabou", yielding in one new publication.

Fuzzy match on fullname The final level of approximation uses n-gram distance between labels of two resources and adds oa:nameSimilarTo4 when the distance is above some threshold. In our data set there is such a statement between the original resource and <http://www.uni4.edu/publications/ins.bib#martha_sabou>, which has "Mart**h**a Sabou" as label. This again added one additional publication, resulting in 9 publications.

Note that we only discussed the *additional search results* in the description above. However, when exploiting the similarities between the authors in the search, we also get duplicate resources for publications for which we apply a similar strategy to combine publication resources.

5 Conclusions

Summary Rough DL is a conservative extension of DL, i.e. an extension of DL with new operators for modeling vague concepts, that does not increase the expressive power of the original language. We show that this language is suitable for reasoning over similarities or equivalences introduced into an ontology through co-reference resolution. Rough DL provides a qualitative way of representing vague concepts, and to reason and query over similarities. By applying Rough DL to openacademia we show that AI techniques can elegantly solve practical problems on the web.

To make the Rough DL version of openacademia robust and efficient for large collections, e.g., crawled on the WWW, the application has initially been restricted to querying. Large scale experiments with Rough DL modeling are planned as future work to evaluate scalability of this theoretically promising framework.

Related Work The related work covers modeling vagueness in ontologies, most prominently in combining fuzzy logic with Semantic Web research, as exemplified in [9]. Some of this work is based on Straccia's paper on fuzzy Description Logics, e.g., [11]. Vagueness of concepts is expressed as a degree of membership. Rough DL advocates a simpler, *qualitative*, approach to domains where there is no way of quantifying membership of the class but well-defined upper and lower approximations. The difference is intrinsically in the type of vagueness of particular concepts. On the querying side, there have also been efforts to integrate

querying over similarities into a standard RDF querying language, e.g., [2]. The language described there, iRDQL, has implicit functionality to query for objects with a certain similarity.

There, however, lies the biggest difference to our approach, which focuses on qualitative modeling of vagueness and querying over similarities. For some domains and particular applications, such as for access to distributed data sources, this approach can be more appropriate. This does not just hold for bibliographic data, but for any data integration where the identity of resources cannot always be established with absolute certainty, and where qualitative querying over similarities can provide a fine-grained access to collections.

Future Work The application of Rough DL to openacademia is a first step towards achieving the full potential of the language. Currently, SeRQL queries are automatically created for narrowing or broadening search results. A next step will be to extend querying to more expressive Rough DL queries, and to integrate Rough DL in the ontology. Together with such an extension of the functionality we will have to undertake a detailed investigation of the scalability of the system, and a qualitative and quantitative analysis of the effects on the querying results in openacademia.

References

1. Niraj Aswani, Kalina Bontcheva, and Hamish Cunningham. Mining Information for Instance Unification. In *ISWC 2006*, 2006.
2. Abraham Bernstein and Christoph Kiefer. Imprecise RDQL: towards generic retrieval in ontologies using similarity joins. In *SAC 2006*, 2006.
3. D.G. Feitelson. On identifying name equivalences in digital libraries. *Information Research*, 9(4):192, 2004.
4. R. Guha and A. Garg. Disambiguating People in Search. *TAP: Building the Semantic Web*, 2003.
5. T.P. Martin. Searching and smushing on the semantic web – challenges for soft computing. In *FLINT 2001 – New Directions in Enhancing the Power of the Internet*, pages 3–8, December 2001.
6. H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. *Advances in Neural Information Processing (NIPS)*, 2003.
7. Z. Pawlak. Rough sets. *Int. Journal of Computer and Information Sciences*, 11:341–356, 1982.
8. Allen Renear and David Dubin. Towards Identity Conditions for Digital Documents. Technical Report UIUCLIS-2003/2+ EPRG, 2003.
9. Elie Sanchez. *Fuzzy Logic and the Semantic Web*. Elsevier, April 5 2006.
10. Stefan Schlobach, Michel Klein, and Linda Peelen. Description logics with approximate definitions: Precise modeling of vague concepts. In *Proc. of the 20th Int. Joint Conf. on Art. Intel., IJCAI 07*, Hyderabad, India, January 2007.
11. Umberto Straccia. Reasoning with fuzzy description logics. *J. of AI Research*, 14:137–166, 2001.
12. Herman J. ter Horst. Completeness, decidability and complexity of entailment for rdf schema and a semantic extension involving the owl vocabulary. *J. of Web Semantics*, 3(2-3):79–115, 2005.