Creating Metadata for the Semantic Web — An Annotation Environment and the Human Factor

Steffen Staab, Alexander Maedche and Siegfried Handschuh
Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
{sst, ama, sha}@aifb.uni-karlsruhe.de
http://www.aifb.uni-karlsruhe.de/WBS

Abstract

Creating metadata by annotating documents is one of the major techniques for putting machine understandable data on the Web. Though there exist many tools for annotating web pages, few of them fully support the creation of semantically interlinked metadata, such as necessary for a truely *Semantic Web*. In this paper, we present an ontology-based annotation environment, *OntoAnnotate*, which offers comprehensive support for the creation of semantically interlinked metadata by human annotators. Based on this environment, we then investigate the human factor of metadata creation. In some experiments, we explore the base line for inter-annotator agreements in a rather typical test setting.

1 Introduction

With the upswing of metadata on the Semantic Web for means like semantic web portals [20], there comes the urgent need for adding semantic metadata to existing web pages such that they are digestible for humans and machines. Though there exists a wide range of sophisticated, even professional, annotation tools (cf. Section 4 on related work), none of the ones that we know of has yet fully exploited the new wealth of possibilities that come with RDF [10] and RDF-Schema [1] as metadata formats. In particular, semantic annotation has so far mostly restricted to describing documents or items in documents *in isolation of each other*. In light of the Semantic Web, what intelligent agents crave for are web pages and items on web pages that are not only described in isolation from each other, but that are also *semantically interlinked*.

We have used semantically interlinked information for gathering knowledge relevant in a particular community of users [20]. The underlying idea was that for that domain a group of users would provide semantic metadata *about the content* of relevant web pages. Thus, our Community Web Portal could present all this knowledge, taking great advantage of semantic structures: personalization by semantic bookmarks ("Fred is interested in RDF research"), conceptual browsing, or the derivation of implicit knowledge (e.g., if John works in a project, which is about XML then he knows something about XML), have been some of the features that thrived by having semantically interlinked information. Similarly, we envision that intelligent agents may profit from semantically interlinked information on the Web in the future.[1]

Building the Community Web Portal we found that there were a number of tricky issues with providing semantic annotation in this manner: First of all, the semantic annotation task does not adhere to a strict template structure, such as Dublin Core to name one of the more sophisticated ones in use. Rather it needs to follow the structure given by schema definitions that may vary with, e.g., domain and purpose. In fact, our intelligent agents rely on *domain ontologies*. Semantic annotations need to be congruent with ontology definitions in order to allow for the advantages we have indicated above. Secondly, semantically interlinked metadata is labor-intensive to produce and, hence, expensive. Therefore duplicate annotation must be avoided. Because semantic annotation is a continuous process in a distributed setting there are several sources for duplication. There is knowledge generated by other annotators. In order to allow for the reuse of their annotations it is important that one does not start from scratch when annotating sources, but that one builds on others efforts (in particular their creation of IDs). Then, there is a multitude of schema descriptions (ontologies) that also change over time to reflect changes in the world. Because manual re-annotation of old web pages seems practically infeasible, one needs an annotation framework that allows to handle ontology creation, mappings and versioning. Thirdly, there is a lack of experience in creating semantically interlinked metadata for web pages. It is not clear how human annotators perform overall and, hence, it is unclear what can be assumed as a baseline for the machine agent. Though there are corresponding investigations for only *indexing* documents, e.g. in library science [12], a corresponding richer assignment of interlinked metadata that takes advantage of the object structures of RDF is lacking. Finally, purely manual annotation is very expensive. Therefore, only very valuable information will be annotated and it is necessary to help the human annotator with his task. What is needed is support for automatic — or at the least, semi-automatic — semantic annotation of web pages.

In this paper, we deal with the first three of the above mentioned issues. Regarding the fourth problem we refer the interested reader to a companion paper [5]. We first (Section 2) present our basic tool for ontology-based semantic annotation and, then, consider the issue of semantic annotation as

[1]Similar projects like WebKB [16], SHOE [8], and, more recently, DAML (`http://www.daml.org`) point in the same direction.

an ongoing process. In particular, interlinkage between objects and evolving metadata schema need to be managed to avoid redundant annotations and re-annotating, respectively. Section 3 deals with an evaluation of hands-on-experiences, exploring an experiment with human subjects. Our objective was to find out about inter-annotator agreement and to come up with some measures about what can be expected from semantic annotation as an input for machine processing.

Before concluding, we discuss related work in the areas of evaluation, evolving schemata and crawling — prerequisite experiences and techniques for useful, semantically interlinked metadata on the Semantic Web.

2 Ontology-based Semantic Annotation

An *ontology* is commonly defined as an explicit, formal specification of a shared conceptualization of an domain of interest. This means that an ontology describes some application-relevant part of the world in a machine-understandable way. The concepts and concept definitions that are part of the ontology have been agreed upon by a community of people who have an interest in the corresponding ontology. The core “ingredients” of an ontology are its set of concepts, its set of properties, and the relationships between the elements of these two sets.

Ontological structures may give additional value to semantic annotations. They allow for additional possibilities on the resulting semantic annotations, such as inferencing or conceptual navigation that we have mentioned before. But also the reference to a commonly agreed set of concepts by itself constitutes an additional value through its normative function. Furthermore, an ontology directs the attention of the annotator to a predefined choice of semantic structures and, hence, gives some guidance about what and how items residing in the documents may be annotated.

Besides of these advantages that ontology-based semantic annotation yields in comparison to “free text metadata generation”, the extended set of capabilities also entails some new problems that need to be solved. In particular, semantic interlinkage between document items incurs the difficulty to adequately manage these interlinkages. Essentially, this means that an ontology-based annotation tool must address the issue of *object identity* and its management across many documents. Also, ontologies may have elaborate definitions of concepts. When their meaning changes, when old concepts need to be erased, or when new concepts come up, the *ontology changes*. Because updating previous annotations is generally too expensive, one must deal with change management of ontologies in relation to their corresponding annotations. Finally, one must prevent redundant annotation which stem from duplicate pages on the web or annotation work done by fellow annotators. Hence, we provide two basic mechanisms for recognizing *document identity*. In the remainder of this section we embed these requirements into a coherent framework.

2.1 OntoAnnotate — The Core Tool

While most annotation tools implicitly subscribe to a particular ontology (e.g., Dublin Core), our tool, OntoAnnotate, makes the relationship between particular ontologies and their parts, i.e. concepts and properties, explicit. OntoAnnotate, presents to the user an interface that dynamically adapts to the given ontology. It has been developed based on our earlier experiences with manual ontology-based semantic annotation that have been described in [5].

As principal language for semantic annotations and ontologies, OntoAnnotate relies on RDF and RDF Schema. RDF Schema can be seen as a language for lightweight ontology descriptions, allowing to define the interlinkage between different concepts (called "classes" in RDFS), properties, and objects (i.e "class members", also called "instances"). To name but a few other possible formats, WebKB uses Conceptual Graphs [16], SHOE employs horn logic rules [8], and we have formerly exploited F-Logic [20]. RDF and RDF Schema, however, provide completely web compatible common denominator that everyone agrees on now. Therefore we have replaced proprietory formats we have used originally.

OntoAnnotate allows for the easy annotation of HTML documents. One may create objects with URIs and relate them to text passages, which are then highlighted. The semantic meaning of the objects and the text passages is given by four semantic categories:

1. **Object identification**: New objects are created by asserting the existence of an object with a unique identifier. The annotation tool supports the creation of object identifiers from text passages.

   This is a mostly syntactic operation, the only semantic consequence is that the set of existing objects is augmented by one.

2. **Object–class relationships**: Each object is assigned to a class of objects by the human annotator. In general, objects may be asserted to belong to multiple classes. To keep the user interface and the evaluation simpler, OntoAnnotate only allows single classification.

3. **Object–attribute relationships**: Each object may be related to attribute values by an attribute. Each attribute value is either a text passage chosen by highlighting or a string typed in by the annotator. For a given object the annotator can only create object–attribute relationships if the object's class definition allows its creation, i.e. if the class definition includes a corresponding attribute.

   An attribute is a property the domain of which is a literal.

4. **Object–object relationships**: Each object may be related to all existing objects (including itself) via an (object) relation. For a given object the annotator can only create object–object relationships if the object's class definition allows its creation, i.e. if the class definition includes a corresponding (object) relation.

   An relation is a property the domain of which is a resource.

Figure 1 shows the screen for navigating the ontology and creating annotations in OntoAnnotate. The left pane displays the document and the right panes show the ontological structures contained in the ontology, namely classes, attributes and relations. In addition, the right pane shows the current *semantic annotation knowledge base*, i.e. existing objects, their classification, object–attribute relationships and object–object relationships created during the semantic annotation.

To illustrate the annotation process with OntoAnnotate, we sketch a small annotation scenario using our tool: Annotation typically starts with identifying a new object. The user provides a new object identifier and selects the appropriate class of this object from a tree view. In our example, the object identifier `RStuder` is typed in and the class FULLPROFESSOR is selected from the ontology. Upon categorization of a new object into a class C, OntoAnnotate shows the possible attributes of C (cf. the attributes ADDRESS, NAME, PHONE, etc. of FULLPROFESSOR in the right upper pane of Figure 1) and the actual attributes of the chosen object (cf. `Karlsruhe, Rudi Studer`, etc. in right upper pane of Figure 1). In addition, one may look at the object relations of C (cf. `affiliation, cooperateWith`, etc. in the right lower pane of Figure 1) and the actual relations of the chosen object. In order to dynamically display the properties of classes and their instances, OntoAnnotate queries the *annotation inference server*. The annotator continues with marking text passages and drags them into empty fields of the attribute table, thereby creating new attributes relationships between the currently chosen object and the currently marked text passage (e.g., between `RStuder` and `studer@aifb.uni-karlsruhe.de` in Figure 1). The annotator may create metadata

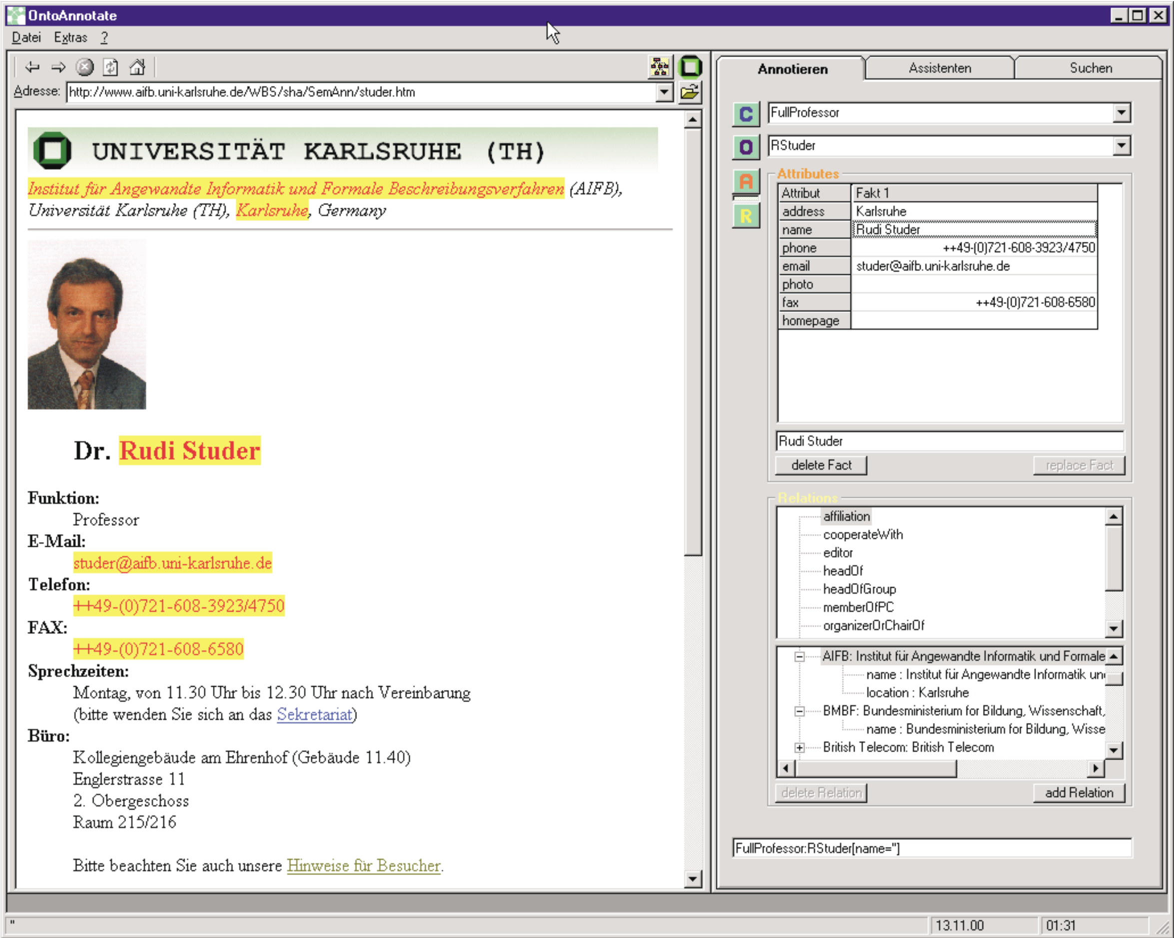

Figure 1: Screenshot of the OntoAnnotate GUI.

describing new object–object relationships by choosing an object relation and, then, either creating a new object on the fly or by choosing one of the objects, pre-selected by OntoAnnotate according to the range restriction of the chosen relation. For instance, the AFFILIATION of a PERSON must be an ORGANIZATION. Therefore, only organizations are offered as potential fillers for the affiliation relation of `RStuder`.

2.2 Object Identity

The first version of OntoAnnotate already relied on ontology structures to guide annotation, but it did not consider annotation as being a process carried through in a complex environment. The general problem stems from the fact that without corresponding tool support, annotators would too often create new objects rather than re-use existing ones. Therefore new properties were not attached to existing objects, but to new entities. In case studies, like the Community Web Portal [20] annotators came up with many different object identifiers for single persons, which made it impossible to combine all the data about these persons.

Considering semantic annotation as a continuous process, we came up with two new requirements:

1. The annotation inferencing server needs to maintain object identifiers during the annotation process.

2. A crawler needs to gather relevant object identifiers for the start of the annotation.

The first requirement is solved by the annotation inference server, by adding objects to and querying objects from the server during actual annotation as described in the previous subsection.

The second requirement has been solved by allowing the annotator to start a focused crawl of RDF facts — covering the document and annotation server, but also relevant parts of the Web — which provides the annotation inference server with an initial set of object identifiers, categories, attributes and relations. Thus, the metadata provided by other annotaters may be used as the starting point that one may contribute additional data to.

Currently, RDF data is comparatively weakly interlinked. Hence, it is sufficient to restrict the focus of the crawl by web server restrictions and depth of the crawl. With more metadata on the Web, one needs to employ more sophisticated techniques in the future.

2.3 Ontology Changes

There exists a tight interlinkage between evolving ontologies and the semantic document annotation. In any realistic application scenario, incoming information that is to be annotated does not only require some more annotating, but also continuous adaptation to new semantic terminology and relationships.

Heflin and Hendler [8] have elaborated in great detail on how ontology revisioning may influence semantic annotations. Therefore, we here only sketch one example revision and its effects:

When an existing class definition is refined, the maintainer of the semantic annotations may explore the objects that belong to this class. He may decide individually or for all objects

- that the objects stay in the class and, hence, the semantic meaning of the annotations is extended by additional semantic constraints;

- that the objects are categorized to belong only to the superclasses of the re-defined class and, hence the semantic meaning of the annotations is reduced by cutting away semantic constraints;

- that the objects are moved to another class.

Along similar lines, other cases of ontology revisions are treated.

The annotation maintainer may explore all the possibilities in the ontology engineering tool, OntoEdit [21] and may define mapping rules to bridge between different ontology revisions. Later on, querying may take advantage of these mappings to also retrieve “old” annotations.

2.4 Document Identity

In order to avoid duplicate annotation, existing semantic annotations of documents should be recognized. Because interesting semantic annotations will eventually refer to external web pages that change, the annotator needs some hints when he encounters a document that has been annotated before, but that may have slightly changed since. Finally, the annotator also needs to recognize that this may be a duplication of another document seen before (e.g. on a mirror site).

For these recognition tasks we provide the following mechanisms: In our local setting we have a document management system where annotated documents and their metadata are stored. OntoAnnotate uses the URI to detect the re-encounter of previously annotated documents and highlights annotations in the old document for the user. Then the user may decide to ignore or even delete the old annotations and create new metadata, he may augment existing data, or he may just be satisfied with what has been annotated before.

In order to recognize that a document has been annotated before, but now appears under a different URI, OntoAnnotate searches in the document management system computing similarity with existing documents by document vector models. If there appear documents the similarity which to the currently viewed document is near 1, then these are indicated to the annotator such that he may check for congruency.

These two techniques for recognizing document identity are very basic, but effective for maintaining document identity in OntoAnnotate, given a dynamic environment such as the Web.

2.5 OntoAnnotate — The Semantic Annotation Environment

The overall annotation environment as outlined in this section is depicted in Figure 2: The core OntoAnnotate is used for viewing web pages and actually providing annotations. It also stores annotated documents in the document management system and adds new metadata to the annotation inference server. The latter is also queried for providing conceptual restrictions given by the ontology. Thus, the annotator's view is restricted to conceptual structures that are congruent with the given ontology.

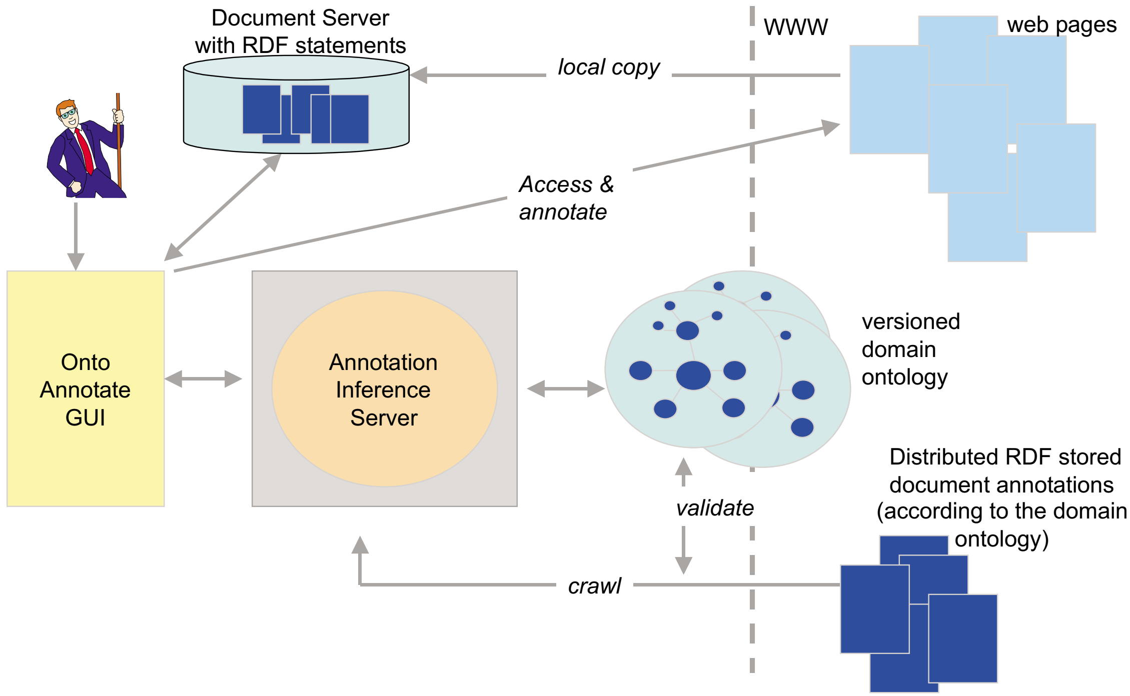

Figure 2: OntoAnnotate — The Semantic Annotation Environment.

The annotation process is started either with an annotation inference server without objects, or the server process is fed with metadata crawled from the Web and the document server. The annotation inference server supports multiple ontologies. Annotations refer to the classes and properties that were used for their creation by namespaces. F-Logic rules are finally used to map between different namespaces, thus allowing to keep track of semantic annnotations (at least to some degree) even when the currently used ontology is replaced by an update.

3 Evaluation of Inter-annotator Agreement

In the last section we have presented our comprehensive annotation environment. In this section we focus on the human annotators. The *human factor* is easily underestimated, but is extremely critical for the manual creation of metadata. In the following we describe an empirical evaluation study of ontology-based semantic annotation. Based on a given ontology and a set of documents, we have analyzed agreement between different humans.

3.1 Evaluation Setting

General Setting. We have evaluated our environment for ontology-based semantic annotation with human subjects performing semantic annotation. In order to determine their inter–annotator agreement, we have undertaken the following experiment: Nine subjects who were undergraduate students in industrial engineering annotated 15 web pages [2] of our institute as part of fulfilling their requirements in a seminar on the "Semantic Web". The domain expertise of the subjects was very sparse. Some of them had some very minor knowledge about the topics and about semantics from introductory courses in computer science, but no prior knowledge of ontologies and semantic annotation. Before doing the actual annotations, subjects received 30 minutes of training. It took about 15 minutes to explain to them the overall goal of semantic annotation and to teach them the basic meaning of the semantic web research community SWRC ontology[3]. The rest of the time was used to acquaint them with the annotation tool. All in all, we have thus expected that the overall achievements could not score very high compared to an expert annotator.

Semantic Annotation Categories. Individual annotation of the 15 test pages led for each annotator to a set of RDF [10] annotated HTML files. From these files we extracted the corresponding annotations as ground facts. In subsection 2.1 we have already introduced the four semantic categories that can be generated within OntoAnnotate. According to these four semantic categories, we distinguished between four different evaluation categories, which are motivated by the varying difficulties they exhibit for the human annotator:

1. The first one only considers the **object identification**. An annotator may choose to use a string as an identifier to denote a new object. These object identifiers play a role similar to that of primary keys in databases. In analogy, we rely on the unique name assumption: Two different identifiers $i_{k,l} \neq i_{m,n}$ are assumed to denote different objects $o_{k,l} \neq o_{m,n}$. In our example, subjects would, e.g., identify an object with identifier `RudiStuder` based on the identifier proposals given by the tool.

[2]The web pages are available at http://ontobroker.semanticweb.org/annotation/SemAnn/

[3]The SWRC ontology models the semantic web research community, its researchers, topics, publications, tools and properties between them. A detailed description of the SWRC ontology and the ontology itself is available at http://ontobroker.semanticweb.org/ontos/swrc.html

2. The second category includes all **object–class relationships**, such as INSTANCE-OF(`RudiStuder`, FULLPROFESSOR) means that `RudiStuder` belongs to the set of FULLPROFESSORs or — precisely speaking — the string "`RudiStuder`" is a unique descriptor for an instance of a FULLPROFESSOR.

3. The third one comprises all **object–attribute relationships**, such as SURNAME-OF(`RudiStuder`, STUDER), which means that the SURNAME-OF the entity the identifier of which is `RudiStuder` is STUDER.

4. The last category is constituted by **object–object relationships**. It includes the relations between two distinct objects, such as MARRIED-TO(`RudiStuder`, `IreneStuder`) or HEADS(`RudiStuder`, `KMResearchGroup`) with their obvious interpretations.

As we will also see in our evaluation in the following, the first one reaches good values based on the proposals by our tool. Object–class assigment is very difficult and results in very low inter–annotator agreement. Attributing seems comparatively easy, where recognizing object–object relationships appears to be the hardest, as it requires elaborate thinking about the denotation of two distinct objects at a rather abstract level.

3.2 Formal Definition of Evaluation Setting

The comparison and evaluation of ontology-based semantic annotation is not this well researched (cf. Section 4 for a detailed comparison of existing work). To the best of our knowledge, no established measure on that we could built did exist. In our semantic annotation scenario we distinguished two different types of measures: On the one hand, we adopt the well-known measures of *precision* and *recall* from the information retrieval community. Whereas these measures denote *perfect agreement*, we additionally define new measures for *sliding agreement*, that take into account string similarity and the sliding scale of the given conceptual structures and compute an inter-annotator accuracy between two annotated document sets.

We now introduce our formal definition for the the semantic annotation scenario. We distinguish between the ontology O (cf. Definition 1) and the semantic annotation knowledge base generated on top (cf. Definition 2).

Definition 1 (Ontology) *An ontology in our framework is a 6-tupel* $\mathcal{O} := (\mathcal{C}, \mathcal{H}, \mathcal{A}, \mathcal{F}_a, \mathcal{R}, \mathcal{F}_r)$ *consisting of a set of classes* $\mathcal{C}$*, which are taxonomically related by the transitive ISA relation* $\mathcal{H}$*,*

($\mathcal{H} \subset \mathcal{C} \times \mathcal{C}$). $\mathcal{A}$ denotes a set of named attributes, $\mathcal{A} \subset \mathcal{C} \times STRING \times STRING$, which allow to relate objects with literals, and $\mathcal{R}$ denotes a set of named relations, $\mathcal{R} \subset \mathcal{C} \times C \times STRING$, which allow to relate objects with each other. A concept $C_i \in \mathcal{C}$ is defined by its place in the taxonomy, and by the attributes and relations that are allowed for its objects. $\mathcal{F}_a$ and $\mathcal{F}_r$ are functions $\mathcal{F}_a : \mathcal{C} \mapsto \mathcal{A}, \mathcal{F}_r : \mathcal{C} \mapsto \mathcal{R}$ that return the attributes and relations that belong to a specific concept C_i, viz. *$\mathcal{F}_a(C_i)$, and $\mathcal{F}_r(C_i)$ respectively. We require that if C_i is a subclass of C_j,* i.e. *$\mathcal{H}(C_i, C_j)$, then $\mathcal{F}_a(C_i) \supseteq \mathcal{F}_a(C_j)$ and $\mathcal{F}_r(C_i) \supseteq \mathcal{F}_r(C_j)$.*

The ontology acts as the conceptual backbone for generating semantic annotations. In our setting we had a subject S_i that generates annotations for a set of documents. The results produced by each of the subjects are defined as Semantic Annotation Knowledge Bases:

Definition 2 (Semantic Annotation Knowledge Base) *The Semantic Annotation Knowledge Base of subject S_i ($SAKB_i$) is a 5-tupel $SAKB_i := (O_i, I_i, c_i, a_i, r_i), i = 1, \ldots, n$ that consists of a set of objects O_i that are uniquely identified by their corresponding literal identifiers I_i ($\subset STRING$). Each object in O_i is assigned to one class $C_j \in \mathcal{C}$ by the* class assignments *of subject S_i,* viz. $c_i \subseteq \{(x, C_j) | x \in O_i,\ C_j \in \mathcal{C}\}$. *Object-attribute relationships are described by the* attribute assignments $a_i \subseteq \{(x, y, z) | x \in I_i,\ y \in \mathcal{A},\ z$ *is a* $STRING\}$ *and object-object relationships by the* relation assignments $r_i \subseteq \{(x, y, z) | x \in I_i,\ y \in \mathcal{R},\ z \in I_i\}$

Note that in our current setting c_i has been restricted by the annotation tool to be functional, i.e. every object could only be assigned to one class.

3.3 Evaluation Measures

3.3.1 Perfect Agreement — Agreement Precision and Agreement Recall

Precision and recall are well known from their definition on the document level. We adopted these two measures for computing the degree to which annotators agree on a set of documents with regard to each of the four semantic annotation categories. Formally, this agreement is computed from the overlap of the elements of two subjects' Semantic Annotation Knowledge Bases $SAKB_i, SAKB_j$. The general notions of *Agreement Precision* (AP) and *Agreement Recall* (AR) are defined based on a pair of sets Q_i, Q_j:

Definition 3 (Agreement-Precision, Agreement-Recall) *Agreement-Precision is defined as* $AP(Q_i, Q_j) := |Q_i \cap Q_j|/|Q_i|$ *and Agreement-Recall defined as* $AR(Q_i, Q_j) := |Q_i \cap Q_j|/|Q_j|$.

Agreement precision and agreement recall are inverses of each other, *i.e.* $AP(Q_i, Q_j) = AR(Q_j, Q_i)$ and $AR(Q_i, Q_j) = AP(Q_j, Q_i)$. Hence, in the following we will only refer to agreement precision, but we will evaluate agreement precision in "both directions". This means, when cross-evaluating annotation results we will evaluate how precisely subject S_i agrees with subject S_j *and vice versa*. Since agreement recall is the inverse of agreement precision this way will also yield all agreement recall numbers. Now, agreement precision for each of the four categories can be simply defined by specifying Q_i and Q_j:

1. **Agreement Precision for Object Identification**: $Q_i := I_i, Q_j := I_j$
2. **Agreement Precision for Class Assignments**: $Q_i := c_i, Q_j := c_j$
3. **Agreement Precision for Attribute Assignments**: $Q_i := a_i, Q_j := a_j$
4. **Agreement-Precision for Relation Assignments**: $Q_i := r_i, Q_j := r_j$

These measures gave us first ideas about the human intra-annotator agreement reachable in our evaluation study using the annotation tool (cf. subsection 3.5). However, the problem is that they lack a sense for the sliding scale of adequacy prevalent in our hierarchical structures. This became obvious especially when comparing the set of object-class relationships C_i (cf. Subsection 3.4). To evaluate the quality of this kind of semantic annotations, we also wanted to add some bonus to annotations that *almost* fitted a annotation in another ontology and, then, to compare annotation schemes on this basis.

3.3.2 Sliding Agreements

In this section we introduce the measures we used to compute the sliding agreements for object identification and class assignment to objects.

Sliding Agreement for Object Identification. In order to compare objects on a string level, one needs a method for comparing and classifying strings that represent the object identifiers in the ontology. One method for judging the similarity between two strings is the *edit distance* formulated by Levenshtein [13]. This is a similarity measures based on the minimum number of token insertions, deletions, and substitutions required to transform one string in another using a dynamic programming algorithm. For example if we calculate the edit distance between the two object identifiers `RudiStuder` and `Rudi Studer` we compute an edit distance of $leven($`RudiStuder`, `Rudi Studer`$) = 1$.

In order to compare two semantic annotation knowledge bases $SAKB_i, SAKB_j$ on a norm scale of $[0,1]$ with 1 for perfect match and near zero for bad match according to the levenshtein measure, we introduce the averaged Identifier Matching Accuracy (IMA) as follows:

(1) $\overline{\text{IMA}}(I_i, I_j) = \frac{1}{|I_i|} \sum_{i_k \in I_i} \text{IMA}(i_k, I_j) \in [0,1]$.

(2) $\text{IMA}(i_k, I_j) = \max_{i_l \in I_j} \frac{1}{1+leven(i_k,i_l)}$.

Sliding Agreement for Class Assignments – Relative Inter-Annotator Agreement. Our new evaluation measure should reflect the distance between the annotation of one annotator to annotations of another annotator. The CMA is a distance based on the hierarchical structure of the ontology and, hence, the resulting conceptual similarity. [15]. Basically, this accuracy measure reaches 100% when both concepts coincide (i.e., their distance $\delta(C_1, C_2)$ in the taxonomy H is 0); it degrades to the extent to which their distance increases; however, this degradation is seen as relative to the extent of their agreement such as given by the distance between their least common superconcept, lcs, and the top concept ROOT. [4]

(3) $\text{CMA}(C_1, C_2) := \dfrac{\delta(\text{lcs}(C_1, C_2), \text{ROOT})}{\delta(\text{lcs}(C_1, C_2), \text{ROOT}) + \delta(C_1, C_2)} \in [0,1]$.

The length of the shortest path $\delta(C_s, C_e)$ between C_s and C_e in the taxonomy H is defined via an auxiliary predicate Path that denotes all the valid paths in H.

(4) $\text{Path}(C_0, \ldots, C_n) :\Leftrightarrow \forall i \in 1 \ldots n : (C_{i-1}, C_i) \in H \cup (C_i, C_{i-1}) \in H$.

(5) $\delta(C_s, C_e) := \min\{n | C_1, ..., C_{n-1} \in C \wedge \text{Path}(C_s, C_1, ..., C_{n-1}, C_e)\}$.

Based on the concept matching accuracy defined above we introduce the **object matching accuracy** OMA. Given two object-class relationships $c_{i,j} = (o_i, C_j)$ and $c_{k,l} = (o_k, C_l)$ OMA is calculated as:

(6) $\text{OMA}(c_{i,j}, c_{k,l}) := \text{CMA}(C_j, C_l) \in [0,1]$.

[4]Multiple inheritance may result in several least common superconcepts for a pair (a, b). Then we continue using the best value for CLA. All the other definitions remain applicable as they are stated here.

The **relative intra-annotator agreement** $\overline{RIAA}$ is based on a weighted OMA. $\overline{RIAA}$ is the averaged accuracy that the object–class annotations of an annotator match against their best counterparts contained in another semantic annotation knowledge base:

(7) $\overline{\text{RIAA}}(c_i, c_k) = \frac{1}{|c_i|} \sum_{c_{i,j} \in c_i} \text{RIAA}(c_{i,j}, c_k))$.

(8) $\text{RIAA}(c_{i,j}, c_k) = \max_{c_{k,l} \in c_l} \{\text{OMA}(c_{i,j}, c_{k,l}\}$.

3.4 An Example Evaluation

Figure 3 depicts an example scenario. In the upper part of the figure, parts of the SWRC ontology are depicted as conceptual backbone for semantic annotation. In the left part of the figure some example we see some annotations done by Annotator 1, in the right part we see some given by Annotator 2. They have produced two different semantic annotation knowledge bases $SAKB_1$ and $SABK_2$, based on the SWRC ontology and the given example webpage.

In the example scenario we can see that the object identifiers `AIFB`, `SteffenStaab` and `PAKM2000` match directly. This results in an agreement-precision for object identifiers computed as $AP(I_1, I_2) = 3/6 = 0.5$. We also see that the object identifier `Alexander Maedche` and the object identifier `AlexanderMaedche` are only similar. Their similarity computes to 0.5, leading to an overall $\overline{IMA}(I_1, I_2)$ of 0.64. The sliding measure $\overline{IMA}$ reflects the fact that there are identifiers that match nearly perfectly.

Looking at the object-class relationships delivers much worse results. One counts only 1 directly matching object-class relationship, namely INSTANCE-OF(`AIFB`, INSTITUTE). So we get an $AP(c_1, c_2)$ of $0.1\overline{6}$. The sliding agreement for object-class relationships recognizes that there are more near hits, namely INSTANCE-OF(`SteffenStaab`, EMPLOYEE) with INSTANCE-OF(`SteffenStaab`, ASSISTANTPROFESSOR) and INSTANCE-OF(`PAKM2000`, CONFERENCE) with INSTANCE-OF(`PAKM2000`, EVENT). We calculate according to the measure defined above an $\overline{RIAA}$ of $0.6\overline{3}$. Additionally we count 0 matching object-attribute relationships and 0 matching object-object relationships, viz. we obtain an $AP(a_1, a_2)$ of 0 and an $AP(r_1, r_2)$ of 0 respectively.

3.5 Cross-evaluation Results

As already mentioned our evaluation is based on the following input parameters: We selected 15 web documents describing actual persons, events, research projects and organizations from our institute.

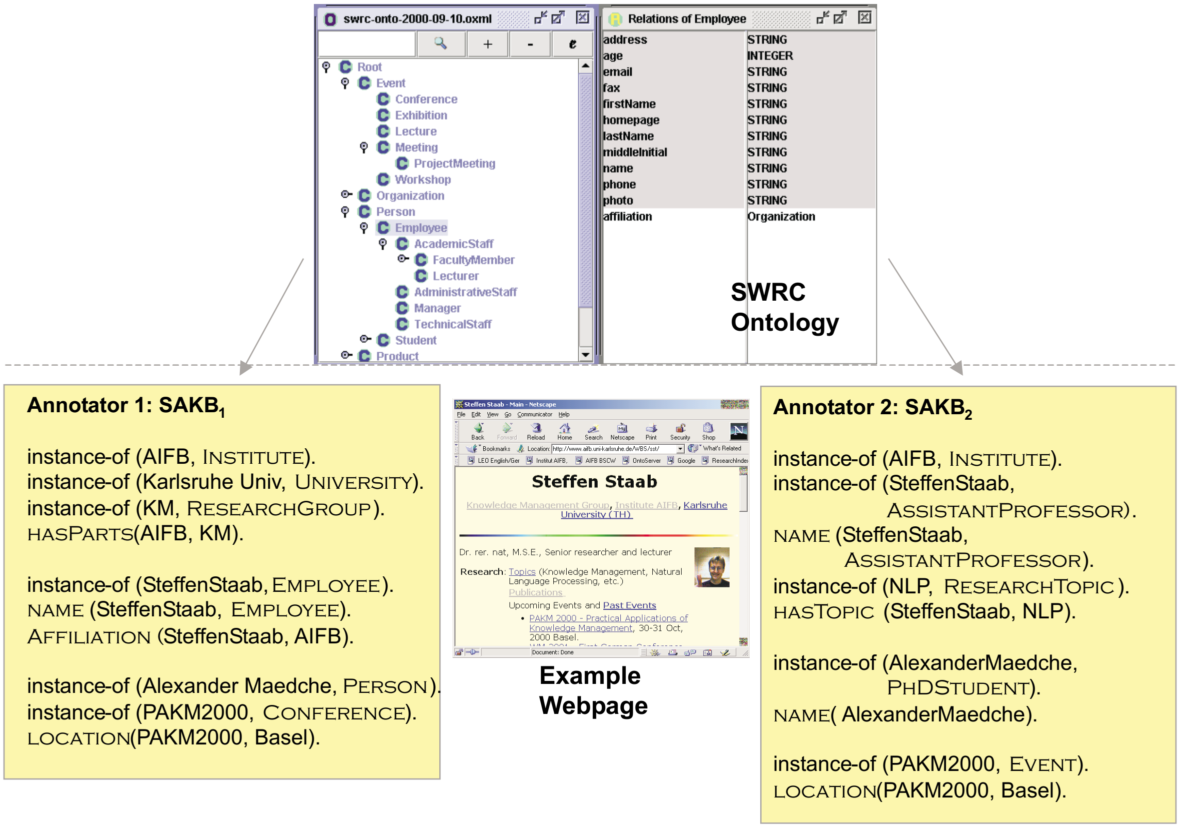

Figure 3: Example Evaluation.

The ontology given to the subjects was the SWRC vocabulary in its current version 2000-10-09 containing 55 classes and 157 attributes and relations. The annotations have been stored in RDF on the web pages. We extracted the annotations from these web pages as ground facts. Our cross-evaluation scenario can be divided into three parts. First, we present some basic statistics we calculated from the semantic annotation knowledge bases. Subsequently, the measures computed using agreement-precision and agreement-recall are explained and interpreted. Additionally, we use $\overline{IMA}$ and $\overline{RIAA}$ to compute the sliding agreement.

Basic statistics. Table 1 shows the basic statistics we obtained in the two phases by counting each semantic annotation category of the semantic annotation knowledge bases. $SAKB_0$ is the semantic knowledge annotation base that has been generated by an expert annotator. It will serve as the gold standard in our evaluation framework.

One may see that the object identifiers with their corresponding object–class relationships have

Subject	$\lvert I_i \rvert$ and $\lvert c_i \rvert$	$\lvert a_i \rvert$	$\lvert r_i \rvert$
$SAKB_0$ (Gold standard)	124	237	183
$SAKB_1$ (anso)	111	206	97
$SAKB_2$ (eryi)	118	162	102
$SAKB_3$ (hela)	82	159	17
$SAKB_4$ (makr)	72	121	29
$SAKB_5$ (mama)	157	293	165
$SAKB_6$ (mari)	97	150	57
$SAKB_7$ (midu)	80	137	46
$SAKB_8$ (stse)	126	226	86
$SAKB_9$ (taso)	104	173	114
mean	107	186	90
standard deviation	26	53	55

Table 1: Basic statistics computed for the generated semantic annotation knowledge bases

an average of 107 elements, with a low standard deviation of 26 elements. Standard deviation of object–attribute and object–object relationships results in a higher value with approx. 50 elements. Some of our students ($SAKB_5, SAKB_8$) have outperformed the gold standard with respect to the basic statistics. In the following we will see what agreement measures are computed based on this 10 given semantic annotation knowledge bases.

Perfect Agreement: Agreement-Precision & Agreement-Recall. Table 2 lists all measures of perfect agreement that we computed in our semantic evaluation study. As highest value for agreement-precision of object identification we obtained 0.75, by comparing the semantic annotation knowledge bases of subject 4 with subject 2. This high value could be obtained by the tool strategy for generating and proposing object identifiers to the users. Agreement-precision of object–class relationships scores much worse, the highest value has been reached with comparing subject 1 with subject 2, namely 0.42. Analyzing Agreement-precision of object–attribute relationships resulted in a maximum reachable value of 0.31 by comparing subject 7 with subject 6.

Figure 4 shows agreement-recall vs. agreement-precision diagrams for matching object identifiers (upper left), matching object–class relationships (upper right), matching object–attribute relationships (lower left) and matching object–object relationships (lower right). Each point in the diagram represents one comparison. We can see from the diagrams that the values obtained for object identifier agreements range between 0.2 and 0.75. The comparison for object–class relationships results range between 0.1 and 0.42. Object–attribute relationships score between 0.05 and 0.31. Object–object relationships score very badly. Some outliers with agreement-precision values of 1 are computed.

	Subject									
Subj.	0	1	2	3	4	5	6	7	8	9
0	-	0.42,0.35	0.47,0.35	0.2,0.19	0.21,0.14	0.45,0.2	0.23,0.15	0.24,0.15	0.31,0.13	0.4,0.17
	-	0.17,0.05	0.16,0.05	0.12,0.0	0.06,0.02	0.12,0.08	0.07,0.07	0.08,0.02	0.13,0.01	0.09,0.02
1	0.47,0.39	-	0.67,0.42	0.41,0.27	0.33,0.15	0.61,0.29	0.38,0.23	0.5,0.25	0.42,0.14	0.65,0.29
	0.2,0.09	-	0.28,0.35	0.24,0.18	0.13,0.0	0.26,0.3	0.16,0.06	0.17,0.1	0.12,0.02	0.22,0.44
2	0.49,0.36	0.63,0.4	-	0.39,0.25	0.46,0.25	0.72,0.25	0.42,0.21	0.48,0.28	0.47,0.19	0.61,0.25
	0.24,0.09	0.35,0.33	-	0.25,0.19	0.17,0.04	0.24,0.16	0.24,0.07	0.2,0.05	0.25,0.06	0.26,0.06
3	0.3,0.28	0.56,0.37	0.56,0.37	-	0.29,0.21	0.46,0.17	0.43,0.26	0.38,0.23	0.34,0.15	0.55,0.23
	0.18,0.0	0.31,1.0	0.25,1.12	-	0.21,0.0	0.16,0.0	0.16,0.0	0.09,0.0	0.09,0.0	0.18,0.0
4	0.36,0.24	0.51,0.24	0.75,0.4	0.33,0.24	-	0.63,0.36	0.5,0.28	0.46,0.22	0.43,0.24	0.54,0.39
	0.12,0.14	0.21,0.0	0.23,0.14	0.27,0.0	-	0.25,0.1	0.19,0.41	0.15,0.0	0.21,0.03	0.36,0.17
5	0.36,0.16	0.43,0.2	0.54,0.19	0.24,0.09	0.29,0.17	-	0.28,0.14	0.36,0.18	0.39,0.11	0.45,0.22
	0.1,0.09	0.18,0.18	0.13,0.1	0.09,0.0	0.1,0.02	-	0.11,0.25	0.14,0.07	0.07,0.05	0.16,0.1
6	0.29,0.19	0.43,0.26	0.51,0.26	0.36,0.22	0.37,0.21	0.45,0.23	-	0.34,0.27	0.33,0.14	0.42,0.23
	0.11,0.21	0.21,0.11	0.26,0.12	0.17,0.0	0.15,0.21	0.21,0.74	-	0.29,0.04	0.14,0.14	0.23,0.44
7	0.38,0.24	0.69,0.35	0.71,0.41	0.39,0.24	0.41,0.2	0.7,0.35	0.41,0.33	-	0.41,0.13	0.69,0.3
	0.14,0.09	0.26,0.22	0.24,0.11	0.11,0.0	0.13,0.0	0.29,0.24	0.31,0.04	-	0.09,0.04	0.24,0.09
8	0.31,0.13	0.37,0.12	0.44,0.18	0.22,0.1	0.25,0.13	0.49,0.14	0.25,0.11	0.26,0.08	-	0.32,0.16
	0.14,0.01	0.11,0.02	0.18,0.07	0.07,0.0	0.11,0.01	0.09,0.09	0.09,0.09	0.06,0.02	-	0.11,0.02
9	0.48,0.2	0.69,0.31	0.69,0.29	0.43,0.18	0.38,0.27	0.67,0.33	0.39,0.21	0.53,0.23	0.38,0.19	-
	0.13,0.03	0.26,0.38	0.24,0.05	0.16,0.0	0.25,0.2	0.28,0.14	0.2,0.22	0.19,0.04	0.14,0.02	-

Table 2: Evaluation Results — Perfect Agreement with AP computed for I_i, c_i, a_i, r_i (order of figures: upper left, upper right, lower left, lower right)

Sliding Agreement. We also computed the sliding agreement measures defined above. Each element of the table contains $\overline{IMA}(I_i, I_j)$ and $\overline{RIAA}(C_i, C_j)$ computed for two semantic annotation knowledge bases, respectively. The values obtained for the identifier matching accuracy did not highly outperform the values we obtained by computing agreement-precision for identifiers. The largest value we received was 0.79 by comparing the knowledge bases of subject 4 with subject 2.

	Subject									
Subj.	0	1	2	3	4	5	6	7	8	9
0	-	0.51,0.9	0.57,0.82	0.32,0.95	0.33,0.7	0.56,0.63	0.34,0.79	0.35,0.71	0.44,0.67	0.5,0.63
1	0.53,0.9	-	0.73,0.77	0.49,0.76	0.41,0.61	0.68,0.6	0.46,0.76	0.58,0.69	0.49,0.55	0.71,0.63
2	0.57,0.82	0.7,0.77	-	0.47,0.79	0.53,0.66	0.77,0.58	0.5,0.72	0.56,0.78	0.54,0.67	0.67,0.62
3	0.4,0.95	0.63,0.76	0.63,0.79	-	0.37,0.8	0.53,0.51	0.49,0.71	0.45,0.75	0.43,0.66	0.6,0.6
4	0.46,0.7	0.6,0.61	0.79,0.66	0.42,0.8	-	0.71,0.71	0.57,0.68	0.54,0.62	0.52,0.76	0.61,0.83
5	0.44,0.63	0.51,0.6	0.6,0.58	0.33,0.51	0.38,0.71	-	0.37,0.65	0.44,0.62	0.48,0.58	0.52,0.66
6	0.38,0.79	0.52,0.76	0.58,0.72	0.43,0.71	0.44,0.68	0.53,0.65	-	0.43,0.84	0.42,0.65	0.5,0.67
7	0.45,0.71	0.77,0.69	0.78,0.78	0.46,0.75	0.48,0.62	0.75,0.62	0.5,0.84	-	0.5,0.57	0.75,0.61
8	0.4,0.67	0.45,0.55	0.51,0.67	0.32,0.66	0.34,0.76	0.58,0.58	0.35,0.65	0.36,0.57	-	0.41,0.67
9	0.55,0.63	0.75,0.63	0.74,0.62	0.49,0.6	0.44,0.83	0.73,0.66	0.47,0.67	0.6,0.61	0.47,0.67	-

Table 3: Evaluation Results — Sliding Agreement Measures: $\overline{IMA}$, $\overline{RIAA}$ (order of figures: left, right)

The values obtained for the relative-inter annotator agreement scored much better than the corresponding agreement-precision values for object–class relationships. The best value of 0.95 was reached by the comparison of semantic annotation knowledge bases generated by subject 0 and sub-

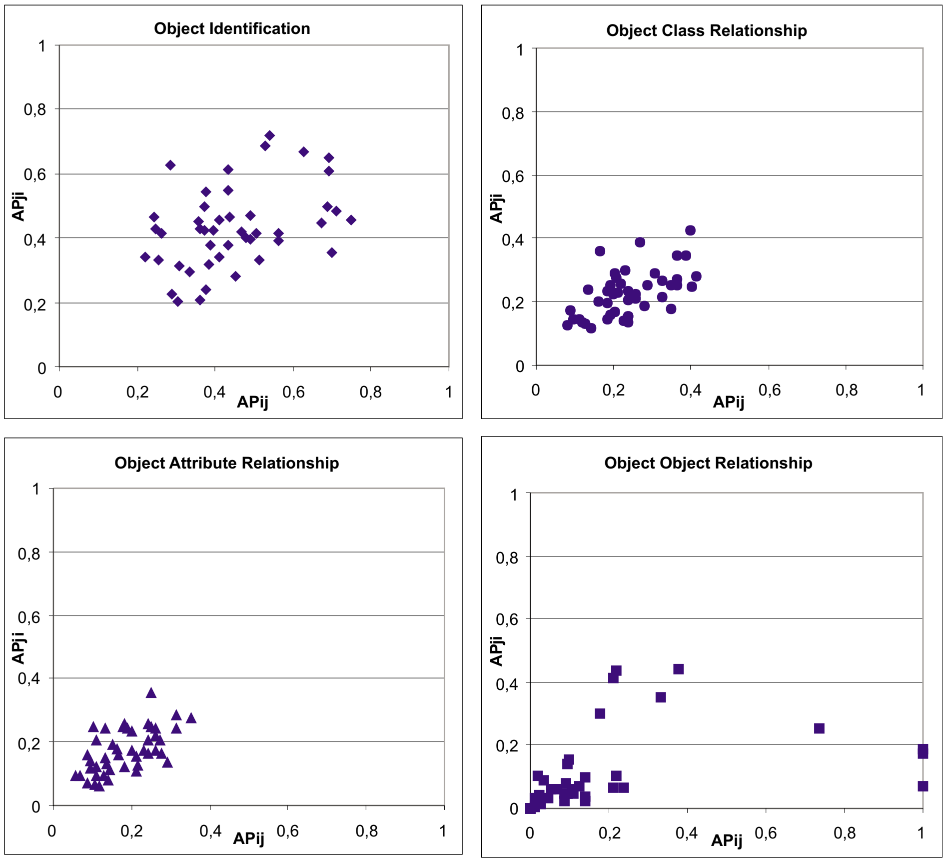

Figure 4: Perfect Agreement

ject 3.

Figure 5 depicts the obtained results graphically. On the left side of figure 5 the resulting comparison values for the identifier matching accuracy are shown. $\overline{IMA}$ ranges around values of 0.5. As shown in the right part of figure 5 the results obtained for computing $\overline{RIAA}$ range between 0.5 and 0.95. The reader may note that $\overline{RIAA}(C_i, C_j)$ is a symmetric measure and therefore returns the same results for $\overline{RIAA}(C_i, C_j)$ and $\overline{RIAA}(C_j, C_i)$.

Overall results. Due to circumstances in our setting, like lack of domain knowledge and no prior experience with the ontologies or with the tool, we believe that this result ranks among the baseline worst cases that will be found in typical semantic annotation settings. Our conjecture is that further training may considerably improve inter-annotator agreement — though we do not expect any numbers for agreement-precision and agreement-recall that range in the vicinity of 100%.

Our evaluation case study goes ahead with several limitations that became obvious during the an-

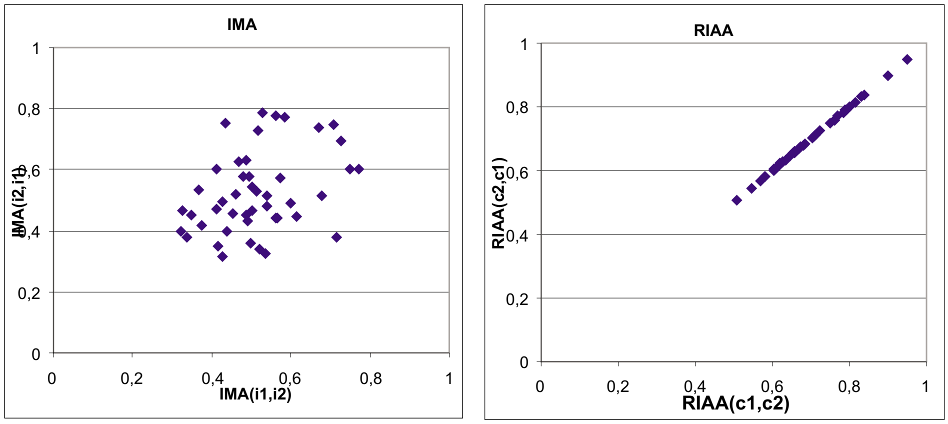

Figure 5: Sliding Agreement

notation experiments. Firstly, the sequence of web pages was given. This may lead to some unwanted similarities of perception. Secondly, the subjects were non-experts but a large part of the domain was about common knowledge. This fact may lead to better results than in more specific domains without common knowledge.

4 Related Work

This paper is motivated by the urgent need for adding metadata to existing web pages in an efficient and flexible manner that takes advantage of the rich possibilities offered by RDF [10] and RDF-Schema [1]. Tools and practices so far have not reflected the new possibilities.

First of all, there is a lack of experience in generating semantically interlinked metadata for web pages. It is not clear how human annotators perform overall. Our work described in this paper extends several existing empirical evaluation studies that have been done in related areas. Second, there are only a few tools that support adding metadata to existing web pages. We will present related work in this area and show how our approach and our implemented tool described in section 2 compares to the existing work. Additionally, our paper introduces semantic annotation as a continuous process. We therefore shortly review existing work in this area.

Related Work on Evaluation. Our evaluation of inter–annotator agreement is an corresponding investigation to studies of consist inter–linking of hypertexts or inter-indexing in library science. On a rough view there is an analogy between indexing a document and identifying ontology objects for documents. There is also an analogy for the creating of links between hypertext nodes compared to creating relations between ontology objects. However, the goals of each approach are not quite comparable and the ontology structures are more complex than hypertext links and indices.

The survey [12] describes the measurement of the extent to which agreement exists among different indexers on the sets of index terms to be assigned to individual documents. The study shows that there is a low level of agreement between the sets of index terms assigned to a document by different indexers. Even the levels of consistency identifiable in the work of a single indexer on a collection of documents are often comparably low. The results from the measurement of inter-linker consistency in hypertext databases as shown in [4] are similar. The work describes an experiment in which the degree of similarity is measured between a number of hypertext databases that share a common set of nodes but whose link-sets have been manually created by different people. In the result the inter–linker consistency is low and varying. The results of inter–indexing and inter–linking studies are comparable with our principal conclusion that high levels of agreement are rarely achieved.

A lot of work on evaluating information extraction systems has been done in the Message Understanding Conferences (MUC). In [11] it is described how the basic evaluation text corpus has been developed in a distributed manner. All contributing sites generated template representations for some specified segment of the 1300 texts. Pairwise combinations of sites were expected to compare overlapping portions of their results and work out any differences that emerged. The authors note that it takes an experienced researcher three days to cover 100 texts and produce good quality template representation for theses texts. However the question, how quality is measured, remains open in their paper. The authors state that their "estimate also finesses the fact that two people will seldom agree on the complete representation which can then be compared, discussed and adjusted as needed."

[18] describes an empirical evaluation of a knowledge acquisition tool with the target of building domain knowledge bases. Military experts have been taken as subjects that had no experience in knowledge acquisition or computer science in general. Evaluation criteria are defined along several dimensions, namely the knowledge-acquisition rate, the ability to find errors, the quality of knowledge entries, the error–covery rate, the retention of skills and the subjective opinion. The results document the ability of these subjects to work on a complex knowledge-entry task and highlight the importance of an effective user interface enhancing the knowledge– acquisition process.

Related Work on Annotation Tools. Koivunen et. al. [9] introduce a framework for categorizing annotation tools distinguishing between a proxy–based and a browser–based approach. The proxy-based approach stores and merges the annotation and therefore preprocesses the annotated documents to be viewable for a standard web-browser. Within the browser–based approach the browser is modified to merge the document with the annotation data just prior to presenting the content to the user.

Many of the annotation tools rely on specialized browsers to offer a better user interface. One of them is *Amaya*. Amaya [7, 23] is a web-browser that acts both as an editor and as a browser. It has been designed at W3C with the primary purpose of being a testbed for experimenting and demonstrating new languages, protocols and formats for the Web. It includes a WYSIWYG editor for HTML and XML. It can publish documents remotely, through the HTTP protocol. It handles Cascading Style Sheets (CSS) and the new MathML language, for representing mathematical expressions. An experiment for including vector graphics into Web documents is also described. Amaya is the primary browser /editor for the annotation approach in [9]. The annotation data itself is exchanged in RDF/XML form to provide other clients access to the annotation database. Currently, however, it does not provide comprehensive support with annotation inference server and crawling.

ComMentor [19] is another browser-based tool as part of the Stanford Integrated Digital Library Project. It manages the meta-information independently of the documents on separate meta-information servers. The research prototype implementation was completed in 1994, the code of the tool is no longer maintained.

ThirdVoice[5] is a commercial product that uses plug-ins to enhance web browsers. This enhancement allows the access to the annotation stored at the ThirdVoice database located on a centralized server from the company. The annotated text parts will appear in the browser as underlined links. These links point to the information on the database that will be presented on the user request in a separate viewer. Most of the annotation stored there seems to be links to further information, so that ThirdVoice is mainly used as a kind of an extended link-list. Along the same lines, *JotBot* [22] follows a browser–based approach that uses Java applets to modify the browsers behavior.

Yawas [3] is an annotation tool that is based on the Document Object Model (DOM) and Dynamic HTML. It codes the annotations into an extended URL format and uses local files similar to bookmark files to store and retrieve the annotations. A modified browser can then transform the URL format into DOM events. Locally stored annotation files can be sent to other users.

[5] http://www.thirdvoice.com

The *CritLink* [24] annotation tool follows the proxy approach. This approach has the advantage that it works with any existing browser. The system is simply used by prefixing the URL with http://crit.org e.g. to see the annotated version of semanticweb.org someone can access the system with the URL http://crit.org/http://semanticweb.org.

The approach closest to *OntoAnnotate* is the *SHOE Knowledge Annotator*[6]. The Knowledge Annotator is a Java program that allows users to mark-up web pages with the SHOE ontology. The SHOE system [14] defines additional tags that can be embedded in the body of HTML pages. In SHOE there is no direct relationship between the new tags and the original text of the page, i.e. SHOE tags are not annotations in a strict sense.

According to the above mentioned classification OntoAnnotate follows the browser-based approach with the exception that it is not developed as an web-browser extension. OntoAnnotate can be regarded as a workbench for semantic annotation of documents using domain-specific ontologies and this enriching HTML pages with semantics that an software agent is capable to automatically process the content of the page and reason about it.

Related Work on Semantic Annotation as a Continuous Process. There is only little research that considers the maintenance of ontologies or more general the maintenance of knowledge bases. In [17] an overview over knowledge maintenance is given. Menzies reviews systems that contribute to different types of knowledge maintenance. The paper analyzes the AI and software engineering literature according to 35 different knowledge maintenance tasks. It concludes that there is no overall strategy that covers all 35 tasks.

The phenomenon of dynamic ontologies has nicely been described in [8]. In their work they discuss the problems associated with managing ontologies in distributed environments such as the web. The underlying representation language is SHOE, a web-based representation language that supports multiple versions of ontologies. Foo [6] has published some initial, theoretical thoughts on ontology revision. Foo outlines the main ideas on the topic of ontology revision and constitutes ontology change as a frontier of knowledge systems research.

[6] http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html

5 Conclusion

This paper presents an approach for creating meta data by annotating web pages. Starting from our ontology-based annotation environment, we have collected experiences in an actual evaluation study. The results provide a baseline that one may consider for further research about automatic annotation tools. The evaluation study we have described was performed with several standard and two original measures. The latter take into account a notion of sliding agreement between meta data — exploiting semantic background knowledge given through the ontology.

Future work will have to start on current studies that have looked at the feasibility of automatic building of knowledge bases from the web (cf. [2]). In our future work, we want to integrate such methods into an even more comprehensive annotation environment — including e.g. the learning of ontologies from web documents [15] and (semi-)automatic ontology-based semantic annotation. The general task of knowledge maintenance, including evolving ontologies and semantic annotation knowledge bases, remains a topic for much further research in the near future.

References

[1] D. Brickley and R.V. Guha. Resource description framework (RDF) schema specification. Technical report, W3C, 1999. W3C Proposed Recommendation. http://www.w3.org/TR/PR-rdf-schema/.

[2] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, to appear.

[3] Laurent Denoue and Laurence Vignollet. An annotation tool for web browsers and its applications to information retrieval. In *In Proceedings of RIAO2000*, Paris, April 2000. http://www.univ-savoie.fr/labos/syscom/Laurent.Denoue/riao2000.doc.

[4] D. Ellis, J. Furner-Hines, and P. Willett. On the measurement of inter-linker consistency and retrieval effectiveness in hypertext databases. In *Proceedings of the 17th annual International ACM SIGIR Conference on Research and Development in Information*, pages 51–60, Berlin, 1994. Springer.

[5] M. Erdmann, A. Maedche, H.-P. Schnurr, and Steffen Staab. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In *P. Buitelaar & K. Hasida (eds). Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*, Luxembourg, August 2000.

[6] N. Foo. Ontology Revision. In *Proceedings of the 3rd International Conference on Conceptual Structures. Springer Lecture Notes in Artificial Intelligence*. Springer, 1995.

[7] R. Guetari, V. Quint, and I. Vatton. Amaya: an Authoring Tool for the Web. In *Maghrebian Conference on Software Engineering and Artificial Intelligence. Tunis, Tunsia*, 1998.

[8] J. Heflin and J. Hendler. Dynamic Ontologies on the Web. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI-2000). Menlo Park, California, AAAI Press*, 2000.

[9] Marja-Riitta Koivunen, Dan Brickley, Jose Kahan, Eric Prud Hommeaux, and Ralph R. Swick. The W3C Collaborative Web Annotation Project ... or how to have fun while building an RDF infrastructure, 2000. http://www.w3.org/2000/02/collaboration/annotation/papers/annotationinfrastructure.

[10] O. Lassila and R. Swick. Resource description framework (RDF) model and syntax specification. Technical report, W3C, 1999. W3C Recommendation. http://www.w3.org/TR/REC-rdf-syntax.

[11] W.G. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. Evaluating an information extraction system. *Journal of Integrated Computer-Aided Engineering*, 1(6), 1994.

[12] L. Leonard. Inter-Indexer Consistence Studies, 1954-1975: A Review of the Literature and Summary of the Study Results. Graduate School of Library Science, University of Illinois. Occasional Papers No.131, 1977.

[13] I. V. Levenshtein. Binary Codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8):707–710, 1966.

[14] S. Luke, L. Spector, D. Rager, and J. Hendler. Ontology-based Web agents. In W. L. Johnson, editor, *Proceedings of the 1st International Conference on Autonomous Agents*, pages 59–66. ACM, 1997.

[15] A. Maedche and S. Staab. Discovering conceptual relations from text. In *ECAI-2000 - European Conference on Artificial Intelligence. Proceedings of the 13th European Conference on Artificial Intelligence*. IOS Press, Amsterdam, 2000.

[16] P. Martin and P. Eklund. Embedding Knowledge in Web Documents. In *Proceedings of the 8th Int. World Wide Web Conf. (WWW'8), Toronto, May 1999*, pages 1403–1419. Elsevier Science B.V., 1999.

[17] T.J. Menzies. Knowledge maintenance: The state of the art. *The Knowledge Engineering Review*, 10(2), 1998.

[18] N. Fridman Noy, W. E. Grosso, and M. A. Musen. Knowledge-acquisition interfaces for domain experts: An empirical evaluation of protege-2000. In *Proceedings of the 12th Internal Conference on Software and Knowledge Engineering. Chicago, USA, July, 5-7, 2000*. KSI, 2000.

[19] Martin Roescheisen, Christian Mogensen, and Terry Winograd. Shared Web Annotations as a Platform for Third-Party Value-Added Information Providers: Architecture, Protocols, and Usage Examples. Technical report stan-cs-tr-97-1582, Computer Science Dept., Stanford University, 1994. http://www-diglib.stanford.edu/diglib/pub/reports/commentor.html.

[20] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, R. Studer, and Y. Sure. Semantic Community Web Portals. In *Proceedings of the 9th World Wide Web Conference (WWW-9), Amsterdam, Netherlands*, 2000.

[21] S. Staab and A. Maedche. Ontology engineering beyond the modeling of concepts and relations. In V.R. Benjamins, A. Gomez-Perez, and N. Guarino, editors, *Proceedings of the ECAI-2000 Workshop on Ontologies and Problem-Solving Methods. Berlin, August 21-22, 2000*, 2000.

[22] Venu Vasudevan and Mark Palmer. On Web Annotations: Promises and Pittfalls of Current Web Infrastucture. In *Proceedings of HICSS'99*, pages 5–8, Maui, Hawaii, January 1999. http://www.objs.com/survey/annotations-hicss.doc.

[23] Irene Vatton. W3C's Amaya 4.0 Editor/Browser. Technical report, W3C, 2000. http://w3c1.inria.fr/Amaya/.

[24] Ka-Ping Yee. CritLink: Better Hyperlinks for the WWW, 1998. http://crit.org/ ping/ht98.html.