

# Ontology supported automatic generation of high-quality semantic metadata<sup>\*</sup>

Ümit Yoldas<sup>1</sup> and Gábor Nagypál<sup>2</sup>

<sup>1</sup> Conemis AG, Karlsruhe, Germany  
yoldas@conemis.com

<sup>2</sup> FZI Research Center for Information Technologies  
at the University of Karlsruhe, Karlsruhe, Germany  
nagypal@fzi.de

**Abstract.** Large amounts of data in modern information systems, such as the World Wide Web, require innovative information retrieval techniques to effectively satisfy users' information need. A promising approach is to exploit document semantics in the IR process. For this purpose, high-quality semantic metadata is needed. This paper introduces a method to automatically create semantic metadata by using ontologically enhanced versions of common information extraction methods, such as named entity recognition and coreference resolution. Furthermore, this work also proposes the application of ontology-specific heuristic rules to further improve the quality of generated metadata. The results of our method was evaluated using a small test collection.

## 1 Introduction

To find relevant documents to a user query, most existing information retrieval (IR) systems merely perform a syntactical comparison between the term-based document representations and the keywords in the query. E.g., if a user initiates a full-text query by typing in the phrase “Semantic Web”, a typical IR system on the Web returns a list of hyperlinks pointing to documents that also contain this string syntactically. Although this method may be sufficient for some applications<sup>3</sup>, it definitely lacks the linguistic and semantic awareness, which becomes increasingly valuable in large information systems and for complex search requests.

In common IR systems, documents are often represented merely as a set of terms — i.e., strings — together with their corresponding frequency measures. In this model, some crucial aspects of natural languages such as synonyms<sup>4</sup> and

---

<sup>\*</sup> This work was partially funded by the VICODI (EU-IST-2001-37534), DIP (FP6-507483), and IMAGINATION (FP6-034626) EU IST projects.

<sup>3</sup> for so-called navigational searches, where the exact keywords in the document are already known for the user

<sup>4</sup> different words that may denote the same things

homonyms<sup>5</sup> are not considered. E.g., from the system’s perspective, the terms “doctor” and “physician” differ completely, although in many documents they have the same meaning.

Apart from this trivial deficiency, several other problems lead to decreased IR performance. According to [1], the major reasons why purely text-based search fails to find some of the relevant documents are the following:

- Abstract concepts: Some high-level, vaguely defined abstract concepts like “World War Two” or “Industrial Revolution” are often not mentioned explicitly in relevant documents. Therefore, most text-based search engines do not consider those documents relevant for queries containing these terms.
- Semantic and temporal relations: No connections can be discovered between the terms “Germany” and “Berlin” or the terms “1990s” and “1994” because non-linguistic relations among concepts are not exploited.

Statistical algorithms can detect coexisting terms in texts, which sometimes (but not always) coincides with semantic relations among terms. Thesaurus-based approaches, such as systems using *WordNet* [2], can exploit some basic linguistic relations. Such approaches cannot, however, handle the problem of indirectly relevant abstract concepts or exploit semantic and temporal relations to find relevant documents.

Ontologies provide an “explicit specification of a conceptualization” [3], and make it possible to define knowledge in a machine-processable form using a formal language such as OWL [4]. Using ontologies, it is possible to exploit semantic and temporal relations to improve IR effectiveness.

Semantic metadata link documents with their relevant ontology instances from the knowledge base (KB). High-quality semantic metadata is a major requirement for any ontology-based information system, including the Semantic Web [5]. Because of the large amount of data in modern information systems, manual or semi-automatic approaches for metadata generation, which rely on significant human input during the annotation process [6–9], are not feasible for most of the applications. It is therefore a very important question how to generate high-quality semantic metadata with as little human effort as possible.

Inspired by the ontology-based IR project VICODI [10], we are currently developing a new ontology-based IR system [1]. This paper describes the metadata generation aspect of this complex system. We present an approach that facilitates the automatic creation of semantic metadata, and provides a framework for the definition of certain ontology-based, domain-specific heuristics. Such heuristics help extend and improve the quality of semantic metadata. To test the claim that such an approach increases the quality of the generated metadata, we evaluated the results of the system using a small test collection.

The structure of the paper is the following. Section 2 gives an overview of our ontology-based metadata generation approach. Section 3 describes our evaluation methodology, and analyzes evaluation results. Section 4 discusses related work, Section 5 concludes the paper and provides some outlook.

---

<sup>5</sup> words with more than one meaning

## 2 Approach

### 2.1 Ontology Formalism

First, some basic requirements are formulated, which have to be met by the ontology formalism in our system. The application domain of our IR system is history and news articles. Therefore, time plays an important role. The ontological structure must provide for temporal restrictions for relation or attribute definitions. These properties are called *temporal relations* and *temporal attributes*, respectively. E.g., it should be possible to define a certain time interval as a validity constraint for the relation `isMemberOf(Steve-Ballmer, Microsoft)`. Moreover, the usual ontological features, such as symmetry and transitivity of relations, and inverse relations, should be supported for temporal relations, too. E.g., we would like to use the temporal transitive `part-of` relation between locations.

As temporal transitive relations are not supported by the current W3C OWL standard [4], an appropriate ontology framework and API is provided by the IR project presented in [1]. Apart from the temporal relations and attributes, our ontology formalism supports the usual ontology modeling constructs, including concepts, instances, relations and attributes<sup>6</sup>. The formalism is implemented using the KAON2 reasoning engine [11], where KAON2 is used as an efficient Datalog engine<sup>7</sup>.

### 2.2 Semantic Metadata Model

Because of performance reasons, we use a metadata model, which is inspired by the common vector space model [12] used in most traditional full-text search engines. This allows us to exploit full-text search engines during the search phase, similarly to [13–15]. The classical vector space model represents documents as a weighted set of terms<sup>8</sup>. Therefore, our model is also based on a weighted set of various model elements.

As was mentioned, semantic metadata links documents with ontology elements. Therefore, our metadata model has a *conceptual part*, which consists of a weighted set of ontology instances<sup>9</sup> (OI). In our system, elements of the conceptual part are termed *weighted ontology instances* (WOI). A WOI contains the URI of the ontology instance, together with its weight, denoting its semantic relevance to the document content.

---

<sup>6</sup> We will sometimes refer to relations and attributes together as “properties”, which is the usual Semantic Web terminology.

<sup>7</sup> In addition, KAON2 also supports OWL-DL reasoning and DL-safe rules, but these features are not used in this work.

<sup>8</sup> Also called as the “bag of words” model.

<sup>9</sup> Although the model does not prohibit using other ontology elements, such as concepts, we use the term ontology instances because in our application scenario the conceptual part includes only instances.

As was also mentioned, time plays a very important role in our application domain. Moreover, from the IR point of view, time has different characteristics from terms or ontology entities. While in the case of terms (and ontology entities) (non-)equality is the only interesting relation, in the case of time, users are mostly interested in other, more complex relations. E.g., if we search for documents about the XX. century we are interested in documents, which are *between* 1901 and 2000. Therefore, our model includes a *temporal part*, which consists of a weighted set of temporal intervals. Actually, for the application domain of history we use fuzzy temporal intervals instead of usual temporal intervals. In this paper, however, we assume for the sake of simplicity that the intervals in the model are all usual time intervals. Details on the fuzzy temporal intervals were reported in [16]. Elements of the temporal part are termed *weighted temporal intervals* (WTI), containing the (fuzzy) temporal interval and its relevance weight.

Finally, it is important to see that for the majority of information systems it is practically impossible to guarantee 100% ontology coverage. I.e., there will be cases, when a relevant concept in the document does not have its respective counterpart in the ontology. For this purpose, we also have a *textual part* in our model, which contains usual strings as its elements. The only difference from the classical full-text vector space model is that our “terms” are not necessary single words. They can be complex phrases, too, such as “Karlsruhe, the German city”. Phrases that are semantically important for the document content, but do not have their counterparts in the ontology, are included in this part of the metadata. Elements of the textual part are termed *weighted terms* (WT), containing the term string and its relevancy weight.

The relevance weight values are between 0.0 (no relevance) and 1.0 (maximum relevance) for all metadata parts.

An example of a possible (partial) metadata of a document describing the causes and consequences of the Russian Revolution is shown in Fig. 1.

```

textual: {"Vladimir Ilich Lenin":1.0, "Attack on the Winter Palace":0.7}
conceptual: { #Lenin:1.0, #Russia:0.8, #Russian-Revolution:1.0 }
temporal: {1917-1920:1.0}

```

**Fig. 1.** Example metadata about the Russian Revolution

### 2.3 Semantic Annotation Steps

We agree with [14] that common *information extraction* (IE) techniques can substantially support the automatized process of metadata generation. We also accept the statement made there that *named entities*<sup>10</sup> (NE) occurring in text

<sup>10</sup> named entities are terms referring to people, organizations or locations; the definition often includes tokens representing dates, percentage numbers etc.

documents constitute a major part of their semantics. Therefore, we start our metadata generation with an IE step, which extracts named entities from the document text. Based on this information, it is possible to generate an initial version of semantic metadata, using the metadata model introduced before. Finally, various ontology-based heuristic rules can be exploited, to extend the initial metadata with relevant ontology entities that are not mentioned in the document text explicitly.

The whole metadata generation process is shown in Fig. 2. In the following, we describe the individual steps in more detail.

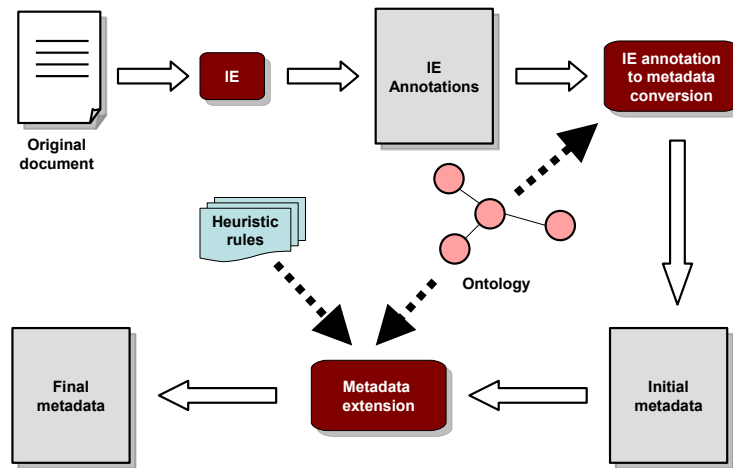


Fig. 2. Steps of the annotation process.

## 2.4 Information Extraction

Classical IE methods comprise some techniques from *natural language processing* (NLP), such as token and sentence splitting or part-of-speech detection. This procedure is sometimes referred to as *shallow parsing* because unlike pure NLP applications, it usually does not include a full (costly) linguistic analysis of the text. The linguistic information obtained by the shallow parsing serves as input to the *named entity recognition* (NER) step.

Detected NEs may also have referring phrases in the text with different notions. These are called *coreferences* and can be distinguished between nominal and pronominal type. An example for a nominal coreference is the term “chief executive officer” referring to the NE “Steve Ballmer” in a document. The “he” reference pointing to “Steve Ballmer” is a pronominal coreference.

To our best knowledge, currently there is no existing IR system which exploits coreference information to create semantic metadata. However, we consider this an important step, since coreference resolution<sup>11</sup> can improve term relevance estimation, as our tests have shown.

With respect to the IE process, we used the established text engineering framework GATE<sup>12</sup>, which includes modules for the NLP, NER and coreference resolution tasks. We used the standard ANNIE components, which are included in the standard GATE installation.

The result of the described IE operations are GATE annotations following a special annotation scheme. They contain detailed linguistic information about each identified term, such as its position within the sentence, part-of-speech information, and a list of its coreferences. These annotations are automatically stored for each document in a relational database for later use during ontology-supported post-processing.

This separation of expensive IE operations from subsequent ontology dependent tasks has some significant advantages. First, linguistic annotations can be generated independently from ontology-lookup operations and thus are independent from any changes in the ontology<sup>13</sup>. Second, different ontology-based heuristics can be applied and tested without complete regeneration of GATE annotations.

## 2.5 Initial Semantic Metadata Generation

In our ontology-based approach, the system must be able to identify appropriate NEs as ontology instances<sup>14</sup>. It is a hard task, because a term in the document can syntactically match many ontology instances. To reduce this ambiguity to a minimum, our implementation follows the “longest match principle”, which is also used by other approaches [17]. According to this principle, always the longest possible text snippet is matched with the ontology instance labels. I.e., we prefer “Bill Gates Foundation” to “Bill Gates”.

Next, linguistic annotations covering an ontology instance are transformed to *ontology annotations* (OIAnnotation). Every OIAnnotation consists of URIs of possibly matching ontology instances (OI), and the number of occurrences of its candidate OIs. The resolved coreferences are taken into account simply by increasing the occurrence counter of the OIAnnotation. E.g., if a pronoun is detected as a coreference to a certain entity, and that entity is known to be an OI, the occurrence counter of the OIAnnotation is increased by one.

The remaining entity annotations are categorized as *term annotations* (TermAnnotation) and *date annotations* (DateAnnotation). Term annotations con-

<sup>11</sup> modern implementations may achieve an F-measure of up to 70 percent

<sup>12</sup> General Architecture for Text Engineering, <http://gate.ac.uk/>

<sup>13</sup> For better coreference recognition, it is sometimes necessary to update the gazetteer lists of GATE based on the ontology labels.

<sup>14</sup> Actually we consider all tokens in the text during this step, not only the text snippets that were identified by GATE as NEs. This is needed because GATE sometimes fails to correctly identify text parts as NEs.

tain the (normalized) terms from the text, together with their occurrence counters; whereas date annotations are special term annotations, where the term text represents a valid date (or time) specification, such as “May 12, 2006” or “today”.

After this step, all annotations are transformed to the initial document metadata, using the model introduced in Section 2.2.

The mapping from linguistic annotations to metadata elements is straightforward. WTIs are generated from date annotations, WTs from term annotations and WOIs from ontology annotations. If an OIAnnotation is ambiguous, i.e., contains more than one possible URIs, WOIs are created for each OI candidate. WOI and WT weights are calculated according to the following logarithmic function:

$$w(x) = \left( \frac{\log(x+1)}{\log(r_{max}+1)} \right)^2 \quad (1)$$

where  $w(x)$  denotes the resulting weight of a new metadata element;  $x$  denotes the occurrence counter of the corresponding annotation element and  $r_{max}$  the largest occurrence counter of all annotation elements for the document. Because the co-occurrence pattern cannot be applied to date specifications, recognized WTIs always get a weight of 1.0 in the current implementation. A more sophisticated weighting scheme for temporal intervals is subject of future work.

Table 1 illustrates the transformation from annotations to an initial document representation (with  $r_{max} = 30$ ).

**Table 1.** Initial metadata generation

<i>Entity</i>	<i>Ann. type</i>	<i>Occurence</i>	<i>Metadata type</i>	<i>Metadata weight</i>
#Bill-Gates	OIAnnotation	30	WOI	1.0
#Microsoft	OIAnnotation	15	WOI	0.65
“Oracle Corporation”	TermAnnotation	7	WT	0.37
“Steve Ballmer”	TermAnnotation	2	WT	0.10
2000-2005	DateAnotation	N/A	WTI	1.0

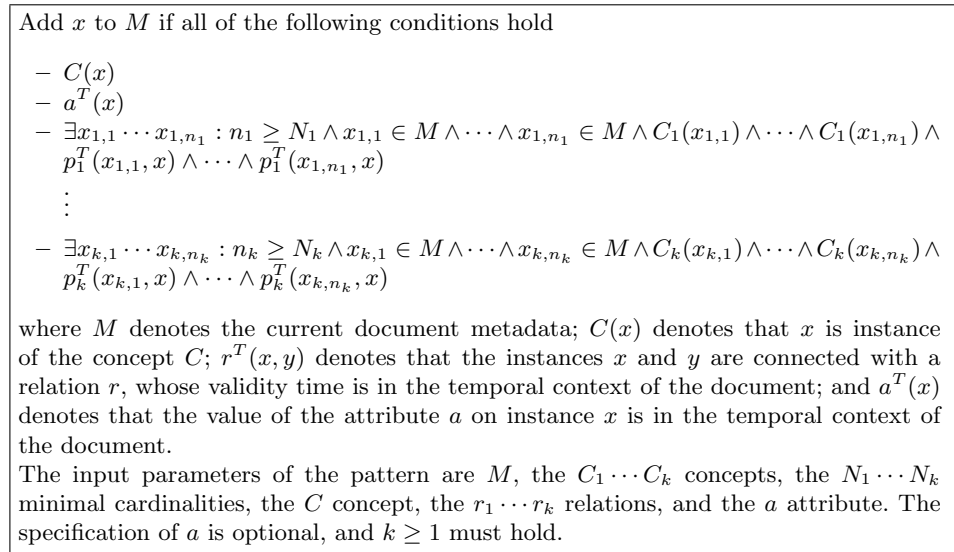
## 2.6 Metadata Extension using Heuristic Rules

The main idea of our approach is that after the initial generation of the document metadata, certain ontology-specific heuristics are used to adapt this metadata to the document’s semantics.

Similar to human readers’ cognitive processing, some basic conclusions are drawn automatically by the system. We achieve this by providing a framework for defining and applying specific rules within the document metadata generation process. A special algorithm iteratively applies these heuristic rules and terminates when no further adaptations can be made to the document metadata.

Our rules follow the pattern shown in Fig. 3. To put it simple, the pattern allows domain experts to specify rules, which add new OIs to the conceptual part of the metadata, if they exist in the temporal context<sup>15</sup> of the document, and if they are connected with other instances in the metadata through relations which are valid in the temporal context of the document. If according to the rule an instance should be added to the metadata, which is already there, only the weight of the instance is adjusted.

Concrete examples of rules following this pattern are shown in Fig. 5(a) and 5(b).



**Fig. 3.** Rule pattern

Our weight calculation scheme for the WOIs introduced by the rules is the following:

$$\bar{w} = \left( \sum_{i=1}^p \frac{w_i}{p} \right) \cdot \prod_{j=p+1}^n \left( 1 + \frac{w_j}{2+p} \right) \quad (2)$$

where  $\bar{w}$  denotes the resulting weight of the new WOI;  $w_1, w_2, \dots, w_p, w_{p+1}, \dots, w_n$  are the weights of all  $n$  WOIs in decreasing order, which are used as input values of the rule with a minimum cardinality of  $p = \sum N_i$ . This means, at least  $p$  WOIs of the current metadata have to meet the rule requirements. The more additional elements are contained in the metadata, the higher the resulting weight of the WOI gets (until a maximum of 1.0).

<sup>15</sup> the temporal context of the document is defined by the temporal part of the metadata

E.g., if a rule needs at least two metadata elements, which must fulfill the rule conditions, but the current metadata contains three such elements with weights 0.7, 0.6 and 0.4, the resulting weight is calculated as

$$\bar{w} = \left( \frac{0.7 + 0.6}{2} \right) \cdot \left( 1 + \frac{0.4}{2 + 2} \right) = 0.65 \cdot 1.1 = 0.715$$

In our system, rules following the pattern from Fig. 3 can be comfortably defined in an XML file. Thus, the rules can be easily adapted without any programming expertise. Using the XML file, additional parameters can be defined, as well. One parameter determines the minimum weight a WOI must have, so that the rule application algorithm uses it. Another parameter is the *weakening factor*, which determines how the weight of the resulting WOI is weakened<sup>16</sup>. This is an important feature, because it ensures the termination of the algorithm (weight changes will monotonically converge to zero).

## 2.7 Metadata Extension Algorithm

Starting with an initial document metadata and some heuristic rules, the metadata extension process is executed. The actual semantic metadata is altered by the following algorithm:

1. Read in the initial document metadata.
2. Read all defined rules from the XML file.
3. Set the current document metadata as the initial metadata.
4. Apply the following steps iteratively
  - (a) Execute all applicable rules on the current metadata.
  - (b) Extend the current metadata with WOIs added by the rules (or adjust the weights of existing WOIs, if they are already there).
  - (c) If the metadata has been modified, restart the iteration.
5. Return the current metadata as the final metadata.

The resulting metadata is the final semantic metadata for our IR system, and stored in the document database for later indexing.

## 3 Evaluation

We compared the quality of metadata generated by our approach with manually generated semantic metadata. This section discusses the evaluation methodology we used and the results of the evaluation.

### 3.1 Document Collection

First, we needed to build a small document collection for the purposes of the evaluation. We chose the domain *IT news* and created a small collection of fifteen documents selected from the *ZDNet* news portal<sup>17</sup>. The selected documents are related to one of the three topics *acquisitions*, *product launches* and *IT fairs*.

<sup>16</sup> The value of the weakening factor is always less than 1.0.

<sup>17</sup> <http://news.zdnet.com/>

### 3.2 Domain Ontology

Using the documents in our document collection, we designed an ontology for the areas *acquisitions*, *product launches* and *IT fairs*. The high-level structure of the resulting ontology is shown in Fig. 4. It is important to see that some of the relations and attributes were defined as temporal (indicated by “T”).

Here we will only briefly introduce some concepts and properties, which we consider helpful to comprehend the example heuristics we will provide later. The ontology itself contains 331 instances, 147 relation instances, 91 temporal relation instances, and 24 temporal attribute values.

Every ontology instance is modeled as direct or indirect instance of the **Thing** concept. An important subconcept of **Perdurant** (a subconcept of **Thing**) is the **Event** class. It is important for the heuristics-based approach because it has many relations to different concepts. Among others, it is related to **Organization** instances (via the relation **hasParticipant**). An example for this relation is: **hasParticipant(Microsoft-launches-BizTalkServer-2004, Microsoft)**.

The temporal attribute **happensDuring** defining the temporal extension of an event, can be exemplified by:  
**happensDuring(PeopleSoft-replaces-CEO, '2004-10-01;2004-10-31')**.

The time interval given by the start date 2004-10-01 and the end date 2004-10-31 marks the interval when the event happened.

### 3.3 Heuristic Rules

Generally, heuristic rules must be carefully aligned with the domain ontology. Otherwise, they may unintentionally disturb the document’s semantics, or may not find the expected relevant concepts.

For our evaluation, we defined fourteen heuristic rules, of which we now introduce two for demonstration purposes. These rules are formulated and briefly explained in Fig. 5(a) and 5(b).

E.g., if a the ontology contains information on the **Peoplesoft-replaces-CEO** event, and the initial semantic metadata contains **Peoplesoft**, **Craig-Conway** and **Dave-Duffield**, **Peoplesoft-replaces-CEO** is added to the metadata according to the *Event rule* (Fig. 5(a)).

Using this extended metadata, the *Process rule* (Fig. 5(b)) adds the information about the high-level concept **Process Oracle’s-takeover-of-Peoplesoft**.

As this example illustrates, heuristic rules can often build on the result of previous rules. This way, semantically related OIs are added stepwise to the initial document representation.

### 3.4 Evaluation Methodology

The evaluation consisted of three steps. First, domain experts were asked to manually define reference document representations for each document in our evaluation corpus. I.e., they were told to select ontology instances from the



Add  $x$  to  $M$  if all of the following conditions hold

- $Event(x)$
- $happensDuring^T(x)$
- $\exists x_1 \dots x_n : n \geq 2 \wedge x_1 \in M \wedge \dots \wedge x_n \in M \wedge Agent(x_1) \wedge \dots \wedge Agent(x_n) \wedge participatesIn(x_1, x) \wedge \dots \wedge participatesIn(x_n, x)$

**Idea:** If at least two agents – i.e., instances of the concepts **Person** or **Organization** – are contained in the current semantic metadata, these agents are related via **participatesIn** to the same event instance, and the temporal context of the document is compatible with the time interval given by the temporal attribute **happensDuring** then the targeted ontology instance of **Event** is considered relevant to document.

(a) Event rule

Add  $x$  to  $M$  if all of the following conditions hold

- $Process(x)$
- $\exists x_1 \dots x_n : n \geq 1 \wedge x_1 \in M \wedge \dots \wedge x_n \in M \wedge Event(x_1) \wedge \dots \wedge Event(x_n) \wedge subPerdurantOf(x_1, x) \wedge \dots \wedge subPerdurantOf(x_n, x)$

**Idea:** At least one ontology instance of the concept **Event** in the metadata leads to the addition of all related process instances (via relation **subPerdurantOf**).

(b) Process rule

**Fig. 5.** Example rules

knowledge base, and assign them to one of five equidistant intervals between 0.0 and 1.0.

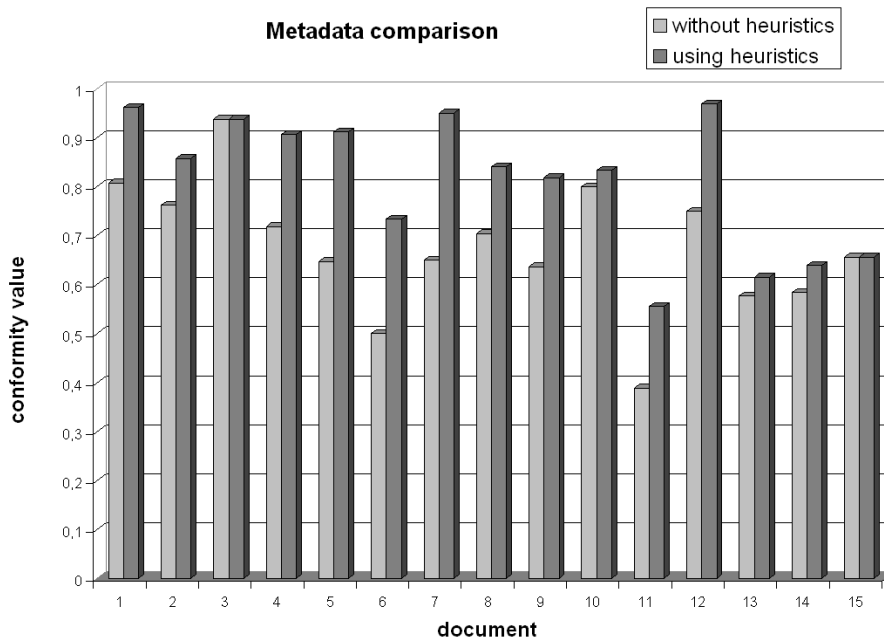
Next, we executed two runs of the system. One run without using any of our heuristics (i.e., only generating the initial metadata representation), and another run with all heuristics activated.

Finally, we compared the generated semantic metadata with the reference metadata defined by the experts, using the following evaluation measure. If the calculated weight of a metadata element lies within the reference interval of that element, a conformity value of 1.0 is assigned to this element. If it is within one interval above or below, a conformity value of 0.5 is assigned. If a calculated element is not contained in the reference metadata, or its weight is not one interval above or below reference weight, the assigned conformity value is 0.0. Finally, for elements that appear only in the reference metadata, but do not appear in the generated metadata, a conformity value of 0.0 is assigned.

By averaging the conformity values for the metadata elements, we obtained average conformity values for each document, which indicate how similar the generated metadata are to the manual (perfect) metadata.

### 3.5 Results

Fig. 6 compares the measured conformity values without heuristics to those where the rules were applied.



**Fig. 6.** Evaluation results

As we can see, almost every document representation was improved by applying the heuristic rules. The average conformity value of the initial document metadata was 0.67, whereas applying the rules lead to an average of 0.81. In none of the cases a decrease in annotation quality was observed. Moreover, no improvement of the semantic metadata could be observed at only two documents (number 3 and 15). In these cases no rules could be applied on the initial metadata. This can be either due to insufficient ontological knowledge<sup>18</sup>, or because the documents would need heuristics which are not described by the existing rule set.

## 4 Related Work

Generating high-quality semantic metadata with the least possible human effort is agreed to be an important step toward the Semantic Web. Thus, there are several projects that aim to support or replace human experts in the metadata generation task.

Approaches such as [6, 8, 18, 9] propose frameworks for manual annotation of metadata. E.g., a GUI<sup>19</sup> application which facilitates the annotation of semantic

<sup>18</sup> WOIs that would be preconditions in rules are not included in the initial metadata

<sup>19</sup> Graphical User Interface

tags is provided by [8]. However, fully manual approaches do not scale well for large information systems.

There are a couple of automatic annotation systems, as well. These include the KIM system [14], the SemTag system [19], and the system of Vallet et al [17]. Unfortunately, they do not exploit the full ontology structure, but use only labels of ontology elements (the KIM system) or exploit only the concept taxonomy in addition to ontology labels (Vallet et al. and SemTag).

S-CREAM [7] uses a semi-automatic annotation approach, which includes a machine learning algorithm. They use the ontology to refine the structure of the semantic metadata, i.e., to find the exact ontological relations between metadata elements. Our approach is different because we concentrate on finding new, only implicitly relevant ontology instances — a task what S-CREAM does not address.

The authors of C-PANKOW [20] propose an advanced annotation and disambiguation system without any machine learning technique. Their approach is to identify correct conceptual entities by measuring statistical information from Google search results. The system uses, however, only syntactical information, i.e., cannot find indirectly mentioned instances.

## 5 Conclusion and Outlook

In this work, we presented a system for automatic ontology-supported generation of semantic metadata, as part of our ontology-based IR system. We showed how common IE methods can be used in combination with a simple lookup on ontology labels to create an initial document representation. On top of that, we propose our framework for heuristic rule definition for semantically extending document metadata.

Our evaluation results verified the thesis that suitably parameterized heuristic rules can indeed significantly improve the quality of semantic metadata. With our selected evaluation corpus, the average conformity value could be increased by 20.9 percent.

As the manual definition and parameterization of adequate heuristic rules for larger ontologies is a quite laborious task, the integration of some automatic techniques may be reasonable. A promising step in this direction is done by [21]. We will consider to integrate the proposed spread-activation approach to our heuristics-based system.

Currently, we are working on a more elaborated temporal information extraction module. Another scheduled improvement is to include a disambiguation step into the metadata generation process. In the current system only new ontology elements can be added, or weights can be adjusted. There is no way, however, to remove apparently irrelevant ontology entities from the semantic metadata — a deficiency, which we would like to address.

## References

1. Nagypál, G.: Improving information retrieval effectiveness by using domain knowledge stored in ontologies. In: *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*. Volume 3762 of *Lecture Notes in Computer Science*. (2005) 780–789
2. Voorhees, E.M.: Using WordNet to disambiguate word sense for text retrieval. In: *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, Pittsburgh, US (1993) 171–180
3. Gruber, T.: A translation approach to portable ontology specifications. *Knowledge Acquisition* **5** (1993) 199–220 the definition of the word "ontology".
4. Dean, M., Schreiber, G.: OWL web ontology language reference. Recommendation, W3C (2004)
5. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* **284** (2001) 34–43
6. Decker, S., Erdmann, M., Fensel, D., Studer, R.: Ontobroker: Ontology based access to distributed and semi-structured information. In Meersman, R., Tari, Z., Stevens, S.M., eds.: *Database Semantics - Semantic Issues in Multimedia Systems, IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics (DS-8)*. Volume 138 of *IFIP Conference Proceedings.*, Kluwer (1999) 351–369
7. Handschuh, S., Staab, S., Ciravegna, F.: S-CREAM - semi-automatic CREation of metadata. In Gómez-Pérez, A., Benjamins, V.R., eds.: *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002*. Volume 2473., Springer (2002) 358–372
8. Hendler, J., Heflin, J.: Searching the web with SHOE. In: *Artificial Intelligence for Web Search. Papers from the AAAI Workshop.*, AAAI Press (2000) 35–40
9. Martin, P., Eklund, P.: Embedding knowledge in web documents. In: *Proceedings of the Eighth International World Wide Web Conference, Toronto, Canada*, Elsevier (1999) 325–341
10. Nagypál, G., Deswarte, R., Oosthoek, J.: Applying the Semantic Web – the VI-CODI experience in creating visual contextualization for history. *Literary and Linguistic Computing* **20** (2005) 327–349
11. Hustadt, U., Motik, B., Sattler, U.: Reducing SHIQ-description logic to disjunctive datalog programs. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*. (2004) 152–162
12. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Communications of the ACM* **18** (1975) 613–620
13. Finin, T., Mayfield, J., Joshi, A., Cost, R.S., Fink, C.: Information retrieval and the Semantic Web. In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*. (2005)
14. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic annotation, indexing, and retrieval. *Journal of Web Semantics* **2** (2005) 49–79
15. Davies, J., Weeks, R.: QuizRDF: Search technology for the Semantic Web. In: *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37 2004)*,. (2004)
16. Nagypál, G., Motik, B.: A fuzzy model for representing uncertain, subjective, and vague temporal knowledge in ontologies. In Meersman, R., Tari, Z., Schmidt, D.C., eds.: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Volume 2888 / 2003 of *Lecture Notes in Computer Science.*, Springer-Verlag (2003) 906 – 923

17. Vallet, D., Fernández, M., Castells, P.: An ontology-based information retrieval model. In: *The Semantic Web: Research and Applications: Second European Semantic Web Conference, ESWC 2005*. Volume 3532 of *Lecture Notes in Computer Science.*, Heraklion, Crete, Greece, Springer (2005) 455–470
18. Kahan, J., Koivunen, M.R., Prud'Hommeaux, E., Swick, R.R.: Annotea: An open RDF infrastructure for shared web annotations. *Computer Networks* **39** (2002) 589–608
19. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In: *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003*, Budapest, Hungary (2003) 178–186
20. Cimiano, P., Ladwig, G., Staab, S.: Gimme' the context: context-driven automatic semantic annotation with C-PANKOW. In Ellis, A., Hagino, T., eds.: *Proceedings of the 14th international conference on World Wide Web, WWW 2005*, Chiba, Japan, ACM (2005) 332–341
21. Rocha, C., Schwabe, D., Aragao, M.P.: A hybrid approach for searching in the semantic web. In: *Proceedings of the 13th international conference on World Wide Web (WWW '04)*, New York, NY, USA, ACM Press (2004) 374–383