EVALUATING ONTOLOGIES WITH RUDIFY

Amanda Hicks and Axel Herold
Berlin-Brandenburgische Akademie der Wissenschaften, Jägerstr. 22/23, 10117 Berlin, Germany

Keywords: Ontology, Ontology development, Ontology evaluation, Rigidity, Type, Role, Wordnet.

Abstract: In this paper we present Rudify, a set of tools designed for the semi-automatic evaluation of ontological meta-properties based on lexical realizations of these meta-properties in natural language. We describe the development of Rudify, provide an evaluation of initial output, and describe how this output can be used in conjunction with OntoClean (Guarino and Welty, 2002) to produce clean ontological hierarchies. In particular we show how a Rudify evaluation of concepts for the meta-property of rigidity can facilitate modelling types and roles.

1 INTRODUCTION

Developing an ontology requires paying especial attention to the hierarchical relations. In particular, taking into consideration certain meta-properties of the concepts modelled in the ontology can help the developer avoid formal contradiction and unsound inheritance of properties (Guarino and Welty, 2004). However, manually determining ontological meta-properties of concepts within large ontologies is time consuming and has been shown to produce a low level of agreement amongst human annotators (Völker et al., 2005). A further difficulty around the annotation of meta-properties is that evaluating the meta-properties of concepts can be difficult for non-ontologists while evaluating technical concepts from a specific domain may be difficult for ontologists who are not trained in this domain.

In this paper we present Rudify, a set of tools for the semi-automatic determination of ontological meta-properties. Rudify has been used for ontology development within the Kyoto project (Herold et al., 2009a; Herold et al., 2009b).

Section 2 of this paper provides an overview of the Kyoto project with particular emphasis on the role of the ontology. Section 3 contains a brief description of OntoClean, a method for evaluating hierarchical relations in an ontology (Guarino and Welty, 2002). Section 4 discusses the meta-property of rigidity and its relation to the type–role distinction. Section 5 discusses the development of Rudify. In section 6 the notion of base concepts is briefly introduced. A set of base concepts was used for the evaluation of the Rudify output (section 7). Finally, section 8 provides specific examples of how Rudify output can be used to “clean up” hierarchical relations within an ontology.

2 THE KYOTO PROJECT

The Kyoto project is a content enabling system that performs deep semantic analysis and searches and that models and shares knowledge across different domains and different language communities. Semantic processors are used for concept and data extraction, and the resulting knowledge can be used across the different linguistic communities. A wiki environment allows domain specialists to maintain the system. Kyoto is currently being targeted toward the environmental domain and will initially accommodate seven languages, namely, English, Dutch, Spanish, Italian, Basque, Chinese, and Japanese. The system depends on an ontology that has been linked to lexical databases (wordnets) for these languages. The role of the ontology is to provide a coherent, stable and unified frame of reference for the interpretation of concepts used in automatic inference. For more information on the Kyoto project see (Vossen et al., 2008) and `http://www.kyoto-project.eu/`.

Kyoto should be able to accommodate, not only a variety of languages and domains of knowledge, but also *changes* in scientific theories as both the world and our knowledge of the world change. We, therefore, require an ontology that is not idiosyncratic, but

Hicks A. and Herold A. (2009).
EVALUATING ONTOLOGIES WITH RUDIFY.
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, pages 5-12
DOI: 10.5220/0002275800050012

rather one that can accommodate

1. a variety of languages and their wordnets,
2. a variety of scientific domains,
3. a variety of research communities,
4. future research in these domains, and
5. can serve as the basis of sound, formal reasoning.

Because the end users will be able to maintain and extend the ontology, it is crucial that the ontology is extended in a clean and consistent manner by non-ontology experts.

With this aim in mind we have developed Rudify. We are using Rudify in conjunction with OntoClean in order to build and maintain a clean ontology. By evaluating the ontological meta-properties of concepts, Rudify facilitates a major step in the construction and maintenance of clean hierarchies.

3 OntoClean

OntoClean (Guarino and Welty, 2002) is a method for evaluating ontological taxonomies. It is based on ontological meta-properties of the concepts that appear in the ontological hierarchy. These meta-properties – namely, rigidity, unity, identity, and dependence – are both highly general and based on philosophical notions. Although OntoClean uses meta-properties to evaluate ontological taxonomies, it is not intended to provide a way of determining the meta-properties themselves. Instead it shows the logical consequences of the users modelling choices, most notably ontological errors that may result in taxonomies after modelling choices have been made (Guarino and Welty, 2004). Rudify helps fill this gap by semi-automatically assigning meta-properties to concepts based on how the concepts are expressed in natural language.

Of the four types of ontological meta-properties used by OntoClean, we focus on rigidity. There are several reasons for this choice. First – and most important in the context of the Kyoto project, the notion of rigidity plays a large role in the distinction between types and roles, since every type is a rigid concept and every role is a non-rigid concept. Second, it is relatively easy to find lexical patterns for rigidity. The lexical patterns are a crucial prerequisite for the programmatical determination of meta-properties as done by Rudify (see section 5). Third, AEON (Völker et al., 2008) also concentrated on rigidity, so there is a basis of comparison of data.

4 RIGIDITY

The notion of rigidity relies on the philosophical notion of essence. An essential concept is one that necessarily holds for all of its instances. For example, being an animal is essential to being a cat since it is impossible for a cat to not be an animal, while being a pet is not essential because any cat can, in theory, roam the streets and, thereby, not be a pet. The idea of essence contains an idea of permanence; Fluffy the cat is an animal for the entire duration of his life. However, the notion of essence is stronger than permanence. While Fluffy can be a pet for his entire life, it nevertheless remains possible for him to cease being a pet.

Armed with the notion of essence, we can now define rigidity. A rigid concept is a concept that is essential to all of its possible instances, i. e., every thing that *could* be a cat *is* in fact a cat. Therefore, “cat” is a rigid concept. However, “pet” is a non-rigid concept since there are individual pets that do not have to be a pet.

Non-rigidity further subdivides into two meta-properties: semi-rigidity and anti-rigidity. Those concepts that are essential to some, but not all, of their instances are semi-rigid, while those that are not essential to any of their instances are anti-rigid. We do not focus on this distinction in our work although Rudify can be used to evaluate these meta-properties as well.

Roles vs. Types

We are currently using Rudify to develop a central ontology, to separate type- and role-hierarchies in OntoWordNet (Gangemi et al., 2003), and also to help the end user keep type- and role-hierarchies in the domain ontology separate. This section provides a discussion of the relation between rigidity and type–role hierarchies.

Types and roles are the two main subdivisions of sortal concepts. A sortal concept is a concept that describes what sort of thing an entity is. For example “cat,” “hurricane,” and “milk” are sortal concepts while “red,” “heavy,” and “singing” are not. In an ontology, sortal concepts are those concepts that carry the meta-property identity (for a discussion of identity, see (Guarino and Welty, 2004)). Furthermore, sortals usually correspond to nouns in natural language. We work on the assumption that the concepts represented in the noun hierarchy of WordNet (Fellbaum, 1998, see also section 6) are sortal terms, since this is generally the case. Types are rigid sortals, while non-rigid sortals are generally roles. Furthermore, roles cannot subsume types.

In order to see that roles should not subsume types, we can consider the following (erroneous) hierarchy:

animal
pet
cat

According to this hierarchy, if Fluffy ceases to be a pet, then Fluffy also ceases to be a cat, which is impossible.

From this last point in conjunction with the above assumption that nouns usually represent sortals, it follows from the OntoClean principles that amongst sortal terms, non-rigid sortals should not subsume rigid sortals. In other words, non-rigid nouns generally should not subsume rigid nouns. There are exceptions to this rule. However, this general conclusion allows us to evaluate concepts only for rigidity and non-rigidity, which in turns saves us the computationally expensive task of evaluating non-rigid terms as either semi- or anti-rigid over large sets of concepts.

5 RUDIFY DEVELOPMENT

The general idea behind Rudify is the assumption that a preferred set of linguistic expressions is used when talking about ontological meta-properties. Thus, one can deduce a concept's meta-properties from the usage of the concept's lexical representation (LR) in natural language. This idea has been developed and programmatically exploited first in the AEON (Automatic Evaluation of ONtologies) project (Völker et al., 2008). AEON was developed for the automatic tagging of existing ontologies in terms of OntoClean meta-properties. The Kyoto project decided to rewrite the software based on the principles published by (Völker et al., 2005) for several reasons: there was no active development of the tool any more and the software was released as a development snapshot only, the web service interface had to be changed due to the maintenance stop of the originally implemented one by Google, and a more flexible input facility was needed instead of the purely OWL based one.

In the following technical description and discussion of Rudify we focus on the meta-property of rigidity as this has been the most important property in the context of the Kyoto project so far.

The first step in the Rudify process is the identification of adequate LRs for the concepts that are to be tagged. Due to polysemeous word forms there is no one to one mapping between concepts and LRs. Also, the actual number of recorded senses for a given LR may vary across lexical databases and across versions of a specific lexical database. The results reported here are based on the English WordNet (Fellbaum, 1998) version 3.0. A further complication are concepts that do not have LRs at all. Typically, this applies mostly for concepts of the top levels of ontologies, although there are some (rare) examples like the missing English antonym for "thirsty" meaning "not thirsty" which constitutes a lexical gap.

A set of linguistic patterns that represent positive or negative evidence for a single meta-property needs to be developed. Each pattern specifies a fixed sequence of word forms. For little inflecting languages like English with relatively fixed word order this approach works reasonably well. Further refinement of the patterns will be needed for languages with more free word ordering. For rigidity, we found only patterns representing evidence *against* rigidity. Thus, the default assumption when tagging for rigidity is that rigidity applies. A concept C is considered non-rigid only if enough evidence against rigidity has been collected for C. Obviously, sparse data for occurrences the LR for C will distort the results and produce a skew in the direction of rigidity.

For rigidity, a typical pattern reads "would make a good X" where X is a slot for a concept's LR. This may be a single token, a multiword or even a complex syntactic phrase (as is frequently the case in Romance languages). Over-generation of patterns is prevented by enumerating and excluding extended patterns. The non-rigid pattern "is no longer (—/a/an) X" over-generates phrases like "there is no longer a *cat* (in the yard/that could catch mice/...)" from which we cannot deduce non-rigidity for "cat."

Another frequent over-generation is found for LRs that occur as part of a more complex compound noun as in "is no longer an *animal shelter*" where *animal* is not an instance of the concept "animal." As the results returned from web search engine are often mere fragments of sentences such instances can only be excluded based on part-of-speech tagging and not based on (chunk) parsing.

Learning to Detect Rigidity

Rudify currently uses 25 different patterns as evidence against rigidity. The results of the web search queries based on these patterns form a feature vector for each LR that is then used for classification, i. e. the mapping from the feature vector to the appropriate rigidity tag. Technically this is a ternary decision between *rigid*, *non-rigid* and *uncertain*.

All classifiers were trained on a hand crafted and hand tagged list of 100 prototypical LRs of which 50 denote rigid concepts and 50 denote non-rigid concepts. They cover a broad range of domains and are recorded as monosemeous (having a single sense) in WordNet.

Four different algorithms have been used for classification:

- decision tree (J48, an implementation ov C4.5)
- multinomial logistic regression
- nearest neighbor with generalization (NNge)
- locally weighted learning, instance based

In evaluating the output we considered the results of all four classifiers and ranked the results according the degree of consensus amongst them (see section 6 for more details).

Both Rudify and AEON rely on the World Wide Web as indexed by Google as the hugest repository of utterances that is accessible to the research community. This is done in order to minimize sparse data effects. We are aware of the theoretical implications that data extracted from Google or other commercial web search engines entails. The most crucial problems are:

- Results are unstable over time. The indexing process is rerun regularly and results retrieved at any given point in time may not be exactly reproducible later.
- The query syntax may be unstable over time and implements boolean searches rather than linguistic searches.
- There are arbitrary limitations of the maximum number of results returned and of the meta-data associated with each result. These may also change over time.
- The data repository is in principle uncontrolled as write access to the World Wide Web and other parts of the Internet is largely unrestricted. Commercial search engines work as additional filters on the raw data with their filter policy often left undocumented and subject to changes as well.

From a linguist's point of view, the first three of these problems are discussed in more detail by (Kilgarriff, 2007).

Rudify now is a highly configurable modular tool with parameter sets developed for English and Dutch. Work is under way for the development of parameter sets for the remaining European languages of the Kyoto consortium (Italian, Spanish, Basque). The software is written in Python and NLTK (Bird et al., 2009) is used as the linguistic backend. Classifier creation, training and application is done using Weka 3 (Witten and Frank, 2005), but can be easily delegated to any software suite capable of manipulating ARFF files. Rudify will be released as free and open source software.

6 BASE CONCEPTS

(Rosch et al., 1976) empirically showed the presence of basic level concepts (BLC) in human cognition. In a conceptual taxonomy, for each concept C its subordinate concepts C_n are typically more specific than C. The increase in specificity is due to at least one added feature for C_n that is compatible with C but allows for discrimination between all C_n. BLCs mark the border between the most general concepts comprising only few features and the most feature rich concepts.

Base concepts (BC) are described by (Izquierdo et al., 2007) as those concepts within a semantically structured lexical data base that "play the most important role" in that data base. This intuitive but vague notion is effectively a rephrase of the BLC. BCs, though, are conceived as a purely computationally derived set based on semantic relations encoded in hierarchical lexical databases. BCs are those concepts that are returned by the following algorithm: for each path p from a leaf node (a node with no hyponym relation to other nodes) up to a root node (a node with no hypernym relation to other nodes) choose the first node C with a local maximum of specific relations to other nodes as a BC. This algorithm can be adapted by defining the set of specific relations (e. g. only hyponymy, all encoded relations including lexical relations) and by defining a minimally required number of subsumed concepts a possible BC must contain. BC sets depend from the specified parameters and the hierarchical structure of the lexical database. Thus, different sets are computed for different versions of WordNet and for other national wordnets. Software and data for computing BCs from WordNet are freely available online at `http://adimen.si.ehu.es/web/BLC`.

WordNet (Fellbaum, 1998) is an electronic lexical database for English. It is organizes words in terms of semantic relations including synonymy ("car"–"automobile"), hyponymy (the relation among general and specific concepts, like "animal" and "cat," that results in hierarchical structures), and meronymy (the part-whole relation, as between "cat" and "claw"). Linking words via such relations results in a huge semantic network.

We have added a set of BCs to the middle level of the Kyoto ontology thereby providing the ontology with a generic set of concepts that can be used for inter-wordnet mappings and wordnet to ontology mappings.

Rudify was evaluated on the set of BCs derived from WordNet 3.0 considering only hypernym relations and with a minimum of 50 subsumed concepts for each BC (BC-50). These parameters result in a set of 297 concepts. Inspecting the BC-50 set we found LRs that are highly unlikely BLCs though they fulfill the formal criteria for BCs. A striking example is "moth." In WordNet, much effort was spent to record a high number of different insects as distinguished concepts thus effectively shifting the basic level downwards in the taxonomic tree. (Tanaka and Taylor, 1991) report on a similar effect of basic level shifts for BLCs that can be shown for experts in their respective domain.

7 EVALUATION OF OUTPUT

We tested Rudify on four different English language data sets:

- 50 region terms (handcrafted by environmental domain specialist)
- 236 Latin species names (selected by environmental domain specialist)
- 201 common species names (selected by environmental domain specialist)
- 297 basic level concepts (BLC-50)

7.1 Domain Specific Terms

Classifiers correctly classified all region terms and all Latin species names as rigid concepts. This holds also for the common English species names with three exceptions: "wildcat" was misclassified as denoting a non-rigid concept by all four classifiers and "wolf" and "apollo" (a butterfly) were mis-classified by all classifiers except NNge. This mis-classification is due to the fact that those LRs are not monosemeously denoting a single concept (a species) but are polysemous and also frequently used in figurative language (examples are taken from our log files):

- "Mount Si High School teacher Kit McCormick is no longer a Wildcat." (generalization from a school mascot to a school member)
- "Also the 400 CORBON is no longer a wildcat." (a handgun)
- "He nearly gave in and became a Wildcat before finally deciding to honor his original commitment to the Ducks." (a football team's (nick)name)
- "For example, the dog is no longer a wolf, and is now a whole seperate species." (example discusses changing relations between concepts over time)
- "For four years, the space agency had been planning, defining, or defending some facet of what led up to and became Apollo."
  (a space mission's name)
- "Others figuring prominently in the county's history were Edward Warren, who established a trading post near what is now Apollo [...]"
  (a geographical name)
- "The patron of the city is now Apollo, god of light, [...]"
  (a Greek deity)

7.2 BLC-50

We classify the Rudify output on the BC-50 set according to the agreement amongst the four classifiers used. We refer to those cases in which all four classifiers reached agreement as *decisive*. Rudify yielded decisive output for 215 BCs. Whenever there is disagreement amongst the classifiers, we refer to this output as *difficult*. There are 82 difficult cases that subdivide into two further cases. When three out of four classifiers reached agreement, we refer to this output as *indecisive*. Rudify yielded indecisive output for 56 BCs. When two classifiers evaluate a term as rigid and two as non-rigid, we refer to this as *undecided*. Rudify is undecided with respect to 26 BCs. These figures are summarized in table 1.

Table 1: General overview of the classification on the BC-50 set.

Rudify output	number of cases
decisive	215
difficult	82
difficult: indecisive	56
difficult: undecided	26

An evaluation of Rudify output for the 215 decisive cases indicates that Rudify produces a high level of accuracy for decisive cases (see table 2). 85 % of the terms evaluated as rigid were correctly evaluated, and 75 % of the terms evaluated as non-rigid are correctly evaluated. Many of the Rudify errors either came from high level concepts, e. g., "artifact" and "unit of measurement," which are ordinarily dealt with manually, or else they dealt with polysemous words, which was an anticipated difficulty (see section 5).

In 3 % of the decisive output we used Rudify to determine whether a concept is rigid or non-rigid, e. g. for "furniture." Since not every concept is ontologically clear cut, and since some concepts lie within areas of ontology in which the alternative theories have not yet been properly worked out (e. g., the ontology of artefacts), we have determined that Rudify can be occasionally helpful in making modelling choices based on the common sense uses of the concepts in language. For these cases the evaluation remains unclear.

For 56 concepts Rudify yielded indecisive output. Exactly 50 % of these cases are incorrect (28 out of 56). For this reason we do not regard the indecisive output to be usable data.

The decisive Rudify output on the BCs within the OWN hierarchy yields five OntoClean errors, if we count the hypernyms, and 22 errors if we count instances of hypernym relations. This is based only on the Rudify output prior to evaluating the correctness of this output, but it gives us an idea of the OntoClean results if we uncritically use Rudify to evaluate concepts in the ontology (for more details, see (Herold et al., 2009b)). In short, Rudify output coupled with the OntoClean methodology provides a useful tool for drawing attention to problems in the backbone hierarchy.

In summary, our evaluation of Rudify output on BCs is that Rudify is successful with respect to the decisive output. It produces decisive output with a relatively high degree of accuracy (83 %) and an overall accuracy on the BC-50 set of 69 % (table 3). Furthermore, Rudify has also proven useful in deciding how to model a few concepts.

Table 2: Overview of the decisively classified BC-50 concepts (215 concepts).

class	evaluation	number of cases	
rigid	incorrect	20	(12 %)
	correct	142	(85 %)
	unclear	5	(3 %)
non-rigid	incorrect	12	(25 %)
	correct	36	(75 %)

Table 3: Summary of evaluation.

classification	number of cases	
correct	206	(69 %)
incorrect	60	(20 %)
undecided	26	(9 %)
decision left to Rudify	5	(2 %)

8 APPLICATION OF OUTPUT

In this section we illustrate with two examples how Rudify results can be used to inform ontology design. The first example uses Rudify independently, the second uses Rudify in conjunction with OntoClean principles.

Example 1

We consider BCs that can reasonably be considered amouts of matter. Amounts of matter are generally referred to by mass nouns; 'milk,' 'mud,' and 'beer' are a few examples. Once again we begin by provisionally modelling the concepts taken from WordNet as the upper level concept "amount-of-matter" into the following hierarchy, which includes rigidity assignments from Rudify. R^+ indicates a rigid concept, R^- indicates a non-rigid concept.

amount of matter
  drug (R^-)
    antibiotic (R^+)
  chemical compound (R^+)
    oil (R^+)
  nutriment (R−)

Using the Rudify data, we can clean up this hierarchy. First we notice that Rudify has evaluated "nutriment" as non-rigid. This indicates that it is probably a role rather than a type. In order to verify this, we refer to the definition taken from WordNet: "a source of materials to nourish the body." That is, the milk in my refrigerator is a nutriment only if it nourishes a body. If you bathe in milk, like Cleopatra, it is a cosmetic. "Nutriment," therefore, is a role that milk can play, so it does not belong in the type hierarchy. We therefore, move it to the role hierarchy as subclass of "amount of matter role." We pause to notice that in this case, the decision was made using Rudify results and human verification of the output. This case does not invoke OntoClean, i. e., there would be no OntoClean errors if "nutriment" were subsumed by "amount of matter." This contrasts with the second example, which yields a formal error within the hierarchy itself.

Example 2

Notice that Rudify evaluates "drug" as non-rigid, and "antibiotic" as rigid. However, the current hierarchy subsumes the rigid concept under the non-rigid concept. This results in a formal error in the hierarchy. Because "drug" and "antibiotic" are both sortal terms, this means a role subsumes a type, which, as we have seen above leads to inconsistency. Consider the antibiotic penicillin. Penicillin is only a drug if it is administered to a patient, but it is always an antibiotic due to its molecular structure. By subsuming "antibiotic" under "drug," the ontology erroneously states that if some amount of penicillin is not administered to a patient, then it is not an antibiotic. The solution then, is to move "drug" out of the type hierarchy and into the role hierarchy. "Drug" then becomes a "substance role," and an antibiotic is subclass of "amount of matter" that can play the role "drug."

Because "chemical compound" and "oil" are both evaluated as rigid we do not need make any changes to this part of the ontology.

The result is the following hierarchy fragments under "amount of matter" and "amount of matter role."

```
amount of matter
  antibiotic
  chemical compound
    oil

mount of matter role
  drug
  nutriment
```

9 CONLUSIONS

We presented Rudify – a system for automatically deriving ontological meta-properties from large collections of text based on the lexical representation of individual concepts in natural language. This approach yields valueable results for use in consistency checking of general large scale ontologies such as the Kyoto core ontology. On the basis of 297 basic concepts derived from the English WordNet 69 % agreement with human judgement could be demonstrated. This closely matches the figures reported by (Völker et al., 2008) for human inter-annotater agreement. For specialized domain terms, agreement was substantially higher: only 3 out of 201 English species terms had been mis-classified.

The evaluation of the results reported here shows potential for further improvement. Word sense disambiguation will increase the accuracy for polysemeous words. First experiments involving hypernyms of LRs in the retrieval of evidence for or against ontological meta-properties give already promising results.

For future reference and stability of the results it will be beneficial to use a controlled linguistic corpus of appropriate size instead of commercial web search engines.

ACKNOWLEDGEMENTS

The development of Rudify and its application to the Kyoto core ontology has been carried out in the EU's 7th framework project *Knowledge Yielding Ontologies for Transition-based Organizations* (Kyoto, grant agreement no. 211423).

The authors would like to thank Christiane Fellbaum for many fruitful discussions and the Kyoto members for their kind collaboration.

REFERENCES

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. The MIT Press.

Gangemi, A., Guarino, N., Masolo, C., and Oltramari, A. (2003). Sweetening wordnet with dolce. *AI Magazine*, 24(3):13–24.

Guarino, N. and Welty, C. (2002). Evaluating ontological decisions with ontoclean. *Communications of the ACM*, 45(2):61–65.

Guarino, N. and Welty, C. (2004). An overview of ontoclean. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, pages 151–172.

Herold, A., Hicks, A., and Rigau, G. (2009a). Central ontology version 1. Kyoto project deliverable D6.2.

Herold, A., Hicks, A., Segers, R., Vossen, P., G. Rigau, G., Agirre, E., Laparra, E., Monachini, M., Toral, A., and Soria, C. (2009b). Wordnets mapped to central ontology version 1. Kyoto project deliverable D6.3.

Izquierdo, R., Suárez, A., and Rigau, G. (2007). Exploring the automatic selection of basic level concepts. In *Proceedings of the International Conference on Recent Advances on Natural Language Processing (RANLP'07)*, Borovetz, Bulgaria.

Kilgarriff, A. (2007). Googleology is bad science. *Computational Linguistics*, 33:147–151.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., and Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitve Psychology*, 8:382–439.

Tanaka, J. W. and Taylor, M. (1991). Object categories and expertise: Is the basic level in the eye of the beholder? *Cognitve Psychology*, 23:457–482.

Völker, J., Vrandecic, D., and Sure, Y. (2005). Automatic evaluation of ontologies (AEON). In *Proceedings of the 4th International Semantic Web Conference (ISWC2005)*, volume 3729 of LNCS, pages 716–731, Berlin/Heidelberg. Springer.

Völker, J., Vrandecic, D., Sure, Y., and Hotho, A. (2008). AEON — an approach to the automatic evaluation of ontologies. *Applied Ontology*, 3(1-2):41–62.

Vossen, P., Agirre, E., Calzolari, N., Fellbaum, C., Hsieh, S., Huang, C., Isahara, H., Kanzaki, K., Marchetti, A., Monachini, M., Neri, F., Raffaelli, R., Rigau, G., and Tescon, M. (2008). Kyoto: A system for mining, structuring and distributing knowledge across languages and cultures. In *Proceedings of LREC 2008*, Marrakech, Morocco.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition.