Supporting Semi-Automatic Semantic Annotation of Multimedia Resources

Jeff Z. Pan and Chris Mellish

Dept. of Computing Science, University of Aberdeen, Aberdeen AB24 3UE, UK
`{jpan,cmellish}@csd.abdn.ac.uk`

Abstract. Ontologies provide an attractive basis for the representation of semantic information to be attached to multimedia information. However, the flexibility available to develop one's own ontology or partially reuse an existing ontology means that human input is unavoidable in the process of creating the annotations. We address the issue of how to use ontology reasoning services and natural language generation to provide presentations of relevant ontology structures for human use.

1 Capturing Intended Meanings with Ontologies

Multimedia resources are complex spatio-temporal signals providing information at several levels of abstraction. How to make the huge amount of multimedia information more easily accessible to automatic processing of multimedia resources by programs is an important issue. One possible solution is to associate multimedia resources with some annotations, with the help of which automatic processes have better understanding of the contents of multimedia resources. However, annotations alone do not establish the semantics of the vocabulary used in the annotations.[1]

One way of giving meaning to annotations is to provide some external agreement on the meaning of a set of information properties. For example, the Dublin Core Metadata Element Set [4] provides 15 'core' information properties, such as 'Title', 'Creator', 'Date', with descriptive semantic definitions (in natural language). One can use these information properties in, e.g., RDF or META tags of HTML. The limitation of the 'external agreement' approach is its inflexibility, i.e., only a limited range of pre-defined information properties can be expressed.

An alternative approach is to use ontologies to specify the meaning of Web resources. *Ontology* is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related. In computer science, ontology is, in general, a 'representation of a shared conceptualisation' of a specific domain [6, 13]. It provides a shared and common *vocabulary*, including important concepts, properties and their definitions, and *constraints*, sometimes referred to as background assumptions regarding the intended meaning of the vocabulary, used in a domain that can be communicated between people and heterogeneous, distributed application systems. The ontology approach is more flexible than the external agreement approach because users can customise vocabulary and constraints in ontologies.

---

[1] For example, an annotation asserting that the object in an image is an elephant does not explain what elephants are.

Example 1. The meaning of a user-defined concept 'adults' can be specified as 'persons whose age are at least 18', where 'persons' can be an atomic concept in an ontology:[2]

$$\text{Adult} \equiv \text{Person} \sqcap \exists age.\texttt{atleast18}.$$

In general, there are at least two advantages of the ontology approach in creating multimedia annotations: (i) It is more flexible; uses can define the vocabulary needed in their domain. (ii) Aspects of the intended meaning of these vocabulary can be explicitly represented. However, while the above advantage (ii) makes it possible for ontology experts to check the intended meaning of terms defined in an ontology, it is usually too hard for ordinary users to check such intended meaning by themselves.

In this paper, we address the issue of how to use ontology reasoning services and natural language generation to provide human-readable presentation of parts of ontologies. We claim this is an important way to support human users who have to create multimedia annotations. The rest of the paper is organised as follows. After a brief introduction of ontology languages (Section 2), we argue that the human beings play an important role in creating multimedia annotations. Based on this observation, we discuss techniques on how to generate natural language to represent axioms in ontologies, with the help of classification (Section 4). We provide a worked example (Section 5) to illustrate our approach before we conclude the paper.

2 Ontology Languages

The technique presented is not restricted to any specific ontology language. As the OWL Web Ontology Language is based on Description Logic, here we will assume that we will use a Description Logic as the ontology language.

Description Logics (DLs) [2] are a family of class-based knowledge representation formalisms, equipped with well-defined model-theoretic semantics [1]. A Description Logic $\mathcal{L}$ consists of an alphabet of distinct concept names (**C**), role names (**R**) and individual (object) names (**I**); together with a set of constructors to construct concept and role descriptions (also called $\mathcal{L}$-*concepts* and $\mathcal{L}$-*roles*, respectively). DLs have a model theoretic semantics, which is defined in terms of interpretations. An *interpretation* (written as $\mathcal{I}$) consists of a *domain* (written as $\Delta^{\mathcal{I}}$) and an *interpretation function* (written as $\cdot^{\mathcal{I}}$), where the domain is a nonempty set of objects and the interpretation function maps each individual name $\texttt{a} \in \mathbf{I}$ to an element $\texttt{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each concept name $\textsf{CN} \in \mathbf{C}$ to a subset $\textsf{CN}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each role name $RN \in \mathbf{R}$ to a binary relation $RN^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function can be extended to give semantics to $\mathcal{L}$-concepts and $\mathcal{L}$-roles (see table 1 for the semantics of concept and role descriptions of OWL DL). Let C, D be $\mathcal{L}$-concepts, C is *satisfiable* iff there exist an interpretation $\mathcal{I}$ s.t. $C^{\mathcal{I}} \neq \emptyset$; C *subsumes* D iff for every interpretation $\mathcal{I}$ we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. A DL knowledge base consists of a set of axioms. Due to the limitation of space, here we only introduce concept inclusion axioms. A concept inclusion axiom is of the form $C \sqsubseteq D$, where C, D are $\mathcal{L}$-concepts. An interpretation $\mathcal{I}$ satisfies $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

---

[2] Please refer to Section 2 for details of the following Description Logics syntax. Note that the Semantic Web standard ontology language does not support customised datatype, such as `atleast18`, which is supported by a datatype extension of OWL DL, called OWL-Eu [10].

Abstract Syntax	DL Syntax	Semantics
Class(A)	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Class(owl:Thing)	$\top$	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
Class(owl:Nothing)	$\bot$	$\bot^{\mathcal{I}} = \emptyset$
intersectionOf($C_1, C_2, \ldots$)	$C_1 \sqcap C_2$	$(C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
unionOf($C_1, C_2, \ldots$)	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
complementOf(C)	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
oneOf($o_1, o_2, \ldots$)	$\{o_1\} \sqcup \{o_2\}$	$(\{o_1\} \sqcup \{o_2\})^{\mathcal{I}} = \{o_1^{\mathcal{I}}, o_2^{\mathcal{I}}\}$
restriction(R someValuesFrom(C))	$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y. \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
restriction(R allValuesFrom(C))	$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y. \langle x, y \rangle \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
restriction(R hasValue(o))	$\exists R.\{o\}$	$(\exists R.\{o\})^{\mathcal{I}} = \{x \mid \langle x, o^{\mathcal{I}} \rangle \in R^{\mathcal{I}}\}$
restriction(R minCardinality(m))	$\geqslant mR$	$(\geqslant mR)^{\mathcal{I}} = \{x \mid \sharp\{y. \langle x, y \rangle \in R^{\mathcal{I}}\} \geq m\}$
restriction(R maxCardinality(m))	$\leqslant mR$	$(\leqslant mR)^{\mathcal{I}} = \{x \mid \sharp\{y. \langle x, y \rangle \in R^{\mathcal{I}}\} \leq m\}$
restriction(T someValuesFrom(u))	$\exists T.u$	$(\exists T.u)^{\mathcal{I}} = \{x \mid \exists t. \langle x, t \rangle \in T^{\mathcal{I}} \wedge t \in u^{\mathbf{D}}\}$
restriction(T allValuesFrom(u))	$\forall T.u$	$(\forall T.u)^{\mathcal{I}} = \{x \mid \exists t. \langle x, t \rangle \in T^{\mathcal{I}} \rightarrow t \in u^{\mathbf{D}}\}$
restriction(T hasValue(w))	$\exists T.\{w\}$	$(\exists T.\{w\})^{\mathcal{I}} = \{x \mid \langle x, w^{\mathbf{D}} \rangle \in T^{\mathcal{I}}\}$
restriction(T minCardinality(m))	$\geqslant mT$	$(\geqslant mT)^{\mathcal{I}} = \{x \mid \sharp\{t \mid \langle x, t \rangle \in T^{\mathcal{I}}\} \geq m\}$
restriction(T maxCardinality(m))	$\leqslant mT$	$(\leqslant mT)^{\mathcal{I}} = \{x \mid \sharp\{t \mid \langle x, t \rangle \in T^{\mathcal{I}}\} \leq m\}$
ObjectProperty(S)	S	$S^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
ObjectProperty(S' inverseOf(S))	S^-	$(S^-)^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
DatatypeProperty(T)	T	$T^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}$

Table 1. OWL concept and property descriptions

In this paper, we use a well known DL reasoning service called *classification*, which is puts *concept names* in their proper place in a taxonomic hierarchy (according to subsumption). As a result of classification, we can obtain the following for a named concept *X*:

1. the named concepts that are *equivalent* to it;[3]
2. the named concepts that it *covers* (minimally subsumes);[4]
3. the named concepts that it *is covered by* (is minimally subsumed by).[5]

If necessary, the basic subsumption relation can be recovered from this information.

3 The Human Element in Annotation

A number of research projects have investigated how annotations for the Semantic Web can be semi-automatically derived for existing Web documents, and these could be adapted for textual documents in general. Techniques include using information extraction (IE) systems that exploit information in particular ontologies [5], providing special

---

[3] X is equivalent to Y iff X subsumes Y and Y subsumes X.

[4] X covers Y iff X subsumes Y, Y is not equivalent to X, and whenever X subsumes Z and Z subsumes Y then Z is equivalent to X or Y.

[5] X is covered by Y iff Y subsumes X, Y is not equivalent to X, and whenever Y subsumes Z and Z subsumes X then Z is equivalent to X or Y.

user interfaces that facilitate human markup of text [14] and using machine learning to infer general extraction patterns from texts that have already been marked up by humans [3]. However, these approaches all assume at least one of the following:

– Ontology terms always map in a straightforward way onto words in a document and vice versa. Yet it is clear that subtle aspects of content are not always signalled in a simple way by specific keywords.
– Generic patterns suffice to detect in natural language relationships from all specific ontologies. This ignores the fact that specific domains can have their own sublanguages and domain-dependent forms of expression (beyond single words).
– Information extraction expertise is available to create specific patterns for every ontology. This is unrealistic, given the specialised nature of IE expertise.
– Human involvement is required, to augment and edit any annotations that can be automatically created and/or create a training corpus for machine learning.

It is clear that, at present, significant human involvement is required for constructing document annotations for *textual* documents. For multimedia resources that are not textual, human involvement is indispensible. In addition, domain experts must make a significant contribution not only to creating annotations, but also to choosing an ontology in the first place (from among possibly a number of existing options) and also tailoring an existing ontology to meet the needs of the new application. But how will these people acquire the expertise to reliably understand and use the distinctions provided by the ontologies they need to work with? A number of tools have been built to enable a user to visualise the structure of an ontology (using mainly graphical approaches), but these tools are mainly designed for knowledge engineers, rather than domain experts or casual ontology users.

4 Natural Language Presentation of Ontologies

We believe that natural language is an important medium to exploit in presenting ontologies (or parts of them) to their users. To understand natural language, a user does not require any specialised training. Natural language is also well equipped to express the complex logical structures that arise in modern ontologies. Thus our research is trying to find ways of exploiting this medium. An ontology takes the form of a set of logical axioms, and so the challenge is to present the material of these axioms in comprehensible way using a language such as English. However, it is important to take on board the fact that the axioms may not come in a form ready for direct realisation in English. The axioms represent one possible way that the material could have been expressed, but there are many other possible ways that this could have been done equally well. For the ontology writer, the choice is arbitrary – because one can rely on reasoning services for ontologies, it is not necessary to worry about which of the many logically equivalent methods of expression to use. This means however that reasoning services must also be be used in natural language generation. The most relevant things to say may not be explicitly stated, but nevertheless may follow from the axioms. So determining the content to be expressed can be seen as a kind of *inference* from the axioms [8].

We now consider how the *classification* reasoning service can be used to provide the basic elements of a natural language answer to two types of questions that people ask when they are familiarising themselves with an ontology or seeking to use it for making distinctions.

4.1 Answering *What is X?*

To answer this question, we first of all classify all concepts, including *X*. The following can then be used to extract the key things to be expressed in an answer:

- Find Z such that X is equivalent to Z (to say "Xs are Zs"). In addition, if there is an axiom directly stating $X \equiv Z$, for some possibly complex concept Z, then this axiom can be stated.
- Find Z such that X is covered by Z (to say "an X is a kind of Z").
- Find Z such that X covers Z (to say "Xs include Zs").

The facts retrieved by these steps are closely related to the "identification", "attributive" and "constituency" rhetorical predicates used in McKeown's "identification schema" [7], and the overall answer could be constructed using the pattern of that schema. It would also be possible to recurse through the Z's introduced in the final step, this giving the effect of the "constituency schema".

4.2 Answering *What is the difference between X and Y?*

For this question, again all concepts should initially be classified, and in addition the concepts $X \sqcup Y$, $X \sqcap \neg Y$ and $Y \sqcap \neg X$ should be named and classified. Useful information for an answer can then be read from the classification results as follows: [6]

- Find Z such that $X \sqcup Y$ is covered by Z (to say "Xs and Ys are both Zs").
- Find minimal Z such that Z subsumes X and it is not the case that Z subsumes Y (to say "Although Xs are Zs, Ys are not necessarily so"). [7]
- Find minimal Z such that Z subsumes Y and it is not the case that Z subsumes X (to say "Although Ys are Zs, Xs are not necessarily so").
- Find maximal Z such that X subsumes Z and it is not the case that Y subsumes Z (to say "Whereas Zs are a kind of X, they are not necessarily a kind of Y").
- Find maximal Z such that Y subsumes Z and it is not the case that X subsumes Z (to say "Whereas Zs are a kind of Y, they are not necessarily a kind of X").
- Find Z such that $X \sqcap \neg Y$ subsumes Z (to say "Whereas Zs are a kind of X, they are never a kind of Y").
- Find Z such that $Y \sqcap \neg X$ subsumes Z (to say "Whereas Zs are a kind of Y, they are never a kind of X").

---

[6] If negations of named concepts are also classified, other possibilities not discussed here include finding Z such that Z subsumes X and $\neg Z$ subsumes Y (to say "Although Xs are Zs, Ys are never Zs").

[7] A good way to find minimal Z subsuming or subsumed by something, and satisfying another condition, is to search through increasingly long paths of the transitive closure of the "covers" relation terminating when the condition is met. Similarly for maximal Z.

Again, ways to organise this material as a coherent text can be found in the natural language generation literature, e.g. McKeown's "compare and contrast" schema and the work of Milosavljevic [9].

5 Worked Example

In this section, we apply our approach to a slightly revised version (for the sake of presentation) of the well know pizza ontology (http://www.co-ode.org/ontologies/pizza/).

$$\text{SpicyPizza} \equiv \text{Pizza} \sqcap \exists has.\text{SpicyTopping} \quad (1)$$
$$\text{VegPizza} \equiv \text{Pizza} \sqcap \forall has.\text{VegTopping} \quad (2)$$
$$\text{AmericanHot} \equiv \text{Pizza} \sqcap \exists has.\text{HopGreenPepperTopping} \sqcap \exists has.\text{DairyTopping} \quad (3)$$
$$\text{HopGreenPepperTopping} \sqsubseteq \text{SpicyTopping} \quad (4)$$
$$\text{Caprina} \equiv \text{Pizza} \sqcap \exists has.\text{TomatoTopping} \quad (5)$$
$$\text{TomatoTopping} \sqsubseteq \text{VegTopping} \quad (6)$$
$$\text{VegTopping} \sqsubseteq \neg(\text{FishTopping} \sqcup \text{MeatTopping} \sqcup \text{DairyTopping}) \quad (7)$$
$$\text{SpicyPizza} \sqsubseteq \text{ForeignPizza} \quad (8)$$
$$\text{VegPizza} \sqsubseteq \text{HealthyPizza} \quad (9)$$
$$\text{SpicyButNotVegPizza} \equiv \text{SpicyPizza} \sqcap \neg\text{VegPizza} \quad (10)$$
$$\text{VegButNotSpicyPizza} \equiv \text{VegPizza} \sqcap \neg\text{SpicyPizza} \quad (11)$$
$$\text{SpicyOrVegPizza} \equiv \text{SpicyPizza} \sqcup \text{VegPizza} \quad (12)$$

Note that, following our approach described in the previous section, we introduce axioms (10)-(12) into the ontology only for the second question below.

| What is AmericanHot? |

The classification service tells us that AmericanHot is covered by SpicyPizza, and there are no named concepts that AmericanHot is equivalent to or covers. There is also one axiom directly stating a complex concept equivalent to AmericanHot. Therefore, our explanation of AmericanHot is as follows: "AmericanHot is a kind of SpicyPizza. AmericanHots are Pizzas that $have$ HopGreenPepperTopping and $have$ DairyTopping".

| What is the difference between SpicyPizza and VegPizza? |

Axioms (10)-(12) are now added to the ontology, but (since they are not concepts in the original ontology) the concepts defined in these axioms are ignored in the selection of concepts to be used in the natural language explanation. The classification service tells us the following:

  i SpicyOrVegPizza is covered by Pizza,
 ii SpicyPizza is covered by ForeignPizza, which does not subsume VegPizza,
iii VegPizza is covered by HealthyPizza, which does not subsume SpicyPizza,
 iv VegPizza covers Caprina, but SpicyPizza does not subsume Caprina,
  v SpicyButNotVegPizza covers AmericanHot.

There are also two axioms directly defining SpicyPizza and VegPizza. Therefore, our explanation of the difference between SpicyPizza and VegPizza is as follows:

> "SpicyPizzas are Pizzas that *have* SpicyTopping. VegPizzas are Pizzas that *have* VegTopping. (i) SpicyPizzas and VegPizzas are both Pizzas. (ii) Although SpicyPizzas are ForeignPizzas, VegPizzas are not necessarily so. (iii) Although VegPizzas are HealthyPizzas, SpicyPizzas are not necessarily so. (iv) Whereas Caprinas are a kind of VegPizza, they are not necessarily a kind of SpicyPizza (v) Whereas AmericanHots are a kind of SpicyPizza, they are never a kind of VegPizza."

6 Conclusion and Outlook

It has been argued that ontologies can be very helpful for multimedia annotations. With the fast development of the research and applications of ontologies, users will soon face a serious issue: given a set of ontologies about the same topic, which one suits their multimedia application best? In this paper, we have addressed the issue of how to use ontology classification reasoning service and natural language generation to provide presentations of relevant ontology structures for ontology users. The motivation of this research is illustrated in the example presented in Section 5. Axioms (1)-(9) are hard to understand for users without a logical background to understand. Although ontology editors can help organising axioms in ontologies and providing graphic representations of axioms to some extent, natural language explanations of the structure underlying the axioms are still necessary. In this paper, we have discussed how to provide explanations for the two questions: "What is X" and "What is the difference between X and Y", which are two useful questions potential users of ontologies would like to ask.

More complex questions (e.g. *What kinds of people have supervisors?*) can be answered by constructing and naming the appropriate complex concept descriptions (here, $X \equiv Person \sqcap \exists supervisor$) and answering one of the above questions for that concept. Such concepts introduce more complex natural language realisation issues than simple named concepts. However, whereas previous work does not address the problem of selecting which facts to present, there are appropriate natural language generation techniques for handling the realisation of complex concepts, as used for instance in the Protégé OWL plugin [11] and older work with DLs [15].

Because there are an infinite number of possible concepts expressible with a given set of atomic concept and property names, it would be impossible to compute subsumption relationships beween all possible pairs of concepts. Thus the classification reasoning service only considers the relationships between named concepts. This means that, for instance, when answering a question *What is a student?* the above approach might be able to say *A student is a kind of person* but it would not be able to respond with *A student has an academic supervisor*, unless the concept $\exists supervisor.Academic$ happens to have been named (and this expansion of the name can be retrieved). In the above, we have artificially named certain extra concepts to be classified, but we cannot know in advance all complex concepts that might be informative in the answer to a question. Further work will address this problem, possibly using approaches from approximate reasoning [12] to generate candidate plausible concepts.

Bibliography

[1] F. Baader, D. L. McGuiness, D. Nardi, and P. Patel-Schneider, editors. *Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press, 2002.

[2] F. Baader and W. Nutt. Basic description logics. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 43–95. Cambridge University Press, 2003.

[3] Fabio Ciravegna and Yorick Wilks. Designing adaptive information extraction for the semantic web in amilcare. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*. IOS Press, Amsterdam, 2003.

[4] DCMI. Dublin Core Metadata Element Set, Version 1.1: Reference Description. DCMI Recommendation, URL `http://dublincore.org/documents/dces/`, June 2003.

[5] John Domingue, Martin Dzbor, and Enrico Motta. Magpie: Supporting browsing and navigation on the semantic web. In *Procs of IUI 2004*, 2004.

[6] T. R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.

[7] Kathleen R. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, Cambridge, 1985.

[8] C. Mellish and X. Sun. Natural language directed inference in the presentation of ontologies. In *Procs of the Tenth European Workshop on Natural Language Geeration*, Aberdeen, Scotland, 2005.

[9] M. Milosavljevic. *Maximising the Coherence of Descriptions via Comparison*. PhD thesis, Macquarie University, 1999.

[10] Jeff Z. Pan and Ian Horrocks. OWL-Eu: Adding Customised Datatypes into OWL. In *Journal of Web Semantics*, 2005. To appear.

[11] Alan Rector, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knoblauch, Robert Stevens, Hai Wang, and Chris Wroe. Owl pizzas: Practical experience of teaching owl-dl: Common errors and common patterns. In E. Motta, editor, *Procs of ECAW 2004*, pages 63–81. Springer LNCS 3257, 2004.

[12] Marco Schaerf and Marco Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74:249–310, 1995.

[13] M. Uschold and M. Gruninger. Ontologies: Principles, Methods and Applications. *The Knowledge Engineering Review*, 1996.

[14] Maria Varges-Vera, Enrico Motta, John Domingue, Mattia Lanzoni, Arthur Stutt, and Fabio Ciravegna. Mnm: Ontology driven semantic sem-automatic and automatic support for semantic markup. In *Procs EKAW2002*. Springer Verlag, 2002.

[15] J. Wagner, J. Rogers, R. Baud, and J-R. Scherrer. Natural language generation of surgical procedures. *Medical Informatics*, 53:175–192, 1999.