Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures

Silvio Peroni, Enrico Motta, and Mathieu d'Aquin

Knowledge Media Institute
The Open University
Milton Keynes, United Kingdom
{s.peroni, e.motta, m.daquin}@open.ac.uk

Abstract. In this paper we address the issue of identifying the concepts in an ontology, which best summarize what the ontology is about. Our approach combines a number of criteria, drawn from cognitive science, network topology, and lexical statistics. In the paper we show two versions of our algorithm, which have been evaluated against the results produced by human experts. We report that the latest version of the algorithm performs very well, exhibiting an excellent degree of correlation with the choices of the experts. While the generation of automatic methods for ontology summarization is an interesting research issue in itself, the work described here also provides a basis for novel approaches to a variety of ontology engineering tasks, including ontology matching, automatic classification, ontology modularization, and ontology evaluation.

Keywords: ontology, semantic web, key concepts, ontology summarization, natural categories, cognitive science.

1. Introduction

The Semantic Web is growing fast and already contains a large amount of data, measured in millions of semantic documents and billions of triples. According to our own estimates, which are based on our experience with the Watson ontology search engine [1], at least seven thousand[1] ontologies[2] exist on the Semantic Web, providing an unprecedented set of resources for developers of semantic applications. Thus, consistently with Mark Stefik's vision of a *knowledge medium* [2], the Semantic Web is rapidly emerging as a large scale platform for publishing and sharing formalized knowledge models. Given this context, for the past two years we have been working on a new generation of knowledge-based applications, which are able to exploit the

---

1 This number refers only to ontologies which are formalised in either OWL, RDFS, or DAML+OIL and are also publicly available on the web.

2 In this context we use the term 'ontology' to refer to a semantic web document, which contains class and relation specifications, rather than simply data about individuals.

Semantic Web as a source of background knowledge, e.g., to provide new solutions to tasks such as ontology matching, or to add semantics to tag spaces [3].

In addition, we have also developed tools, such as the Watson Plug-in, which exposes the functionalities provided by Watson within ontology engineering editors, such as Protégé (http://protege.stanford.edu) and the NeOn Toolkit (http://neon-toolkit.org), thus making it possible for ontology developers to locate relevant semantic web entities, and integrate them with the ontology under construction.

While the vision of a large scale reuse of semantic resources available on the web is in principle very exciting, in reality the current level of tool support for the process of ontology development by reuse is rather limited. For example, while the aforementioned Watson Plug-in makes it possible to locate entities on the Semantic Web and import them into an ontology, it actually provides only limited support for navigating and making sense of the ontologies in which these entities reside. Indeed, a key problem faced by an ontology engineer when considering the reuse of an ontology is *ontology understanding*: how to make sense speedily of the content and organization of an ontology, in order to make decisions about the suitability of the ontology in question for the current ontology engineering development project.

A number of people have partially tackled this problem from different angles. For example, the ontology engineering environments available today, such as Protégé, TopBraid Composer (http://www.topbraidcomposer.com/), or The NeOn Toolkit, all provide functionalities for exploring and visualizing an ontology, to facilitate ontology understanding. Nevertheless, formal evaluations of these tools [4] indicate that these environments do not actually do a particularly good job in helping a user to deal with multiple ontologies, to make sense of an ontology, or in general to develop ontologies by reuse. In particular the aforementioned study reported on the lack of *abstraction mechanisms* in these tools, both at the micro-level (notation) and at the macro-level (providing high level ontology summaries).

In this paper we focus on the latter problem and we present an approach to identifying the *key concepts* in an ontology, to generate a meaningful snapshot of an ontology and facilitate the process of ontology understanding. In contrast with other approaches to ontology summarization [5, 6] our work integrates criteria from both cognitive science, lexical statistics, and graph analysis, to try and come up with the same kind of summaries as human experts.

We will start the discussion in the next section by illustrating both the high-level criteria, which inform our approach, and their initial computational realization. We will then discuss the results obtained from an empirical evaluation of this initial version of our method, which unfortunately showed a low degree of correlation with the choices made by human experts. This negative result led to a revision of our algorithm, which is described in section 3. Among other things, this new version introduces an additional criterion, which attempts to estimate the *popularity,* determined using lexical statistics, of a concept in the ontology. As discussed in section 3.3, the revised version of the algorithm shows an excellent degree of correlation with human experts. Finally, in sections 4-6, we discuss related work, reiterate the key contributions of this work, and outline a number of new opportunities for research and development made possible by it.

2. Our Approach

Our aim is to design a method that, given an ontology and an integer *n*, extracts the *n* concepts, which can be considered as 'best descriptors' of the ontology: the *key concepts*. Obviously there is no formal definition of what is a key concept and, especially if we take a task-independent stance, it is unlikely that such a formal definition can be produced. For this reason, our work is empirically grounded and specifically our goal is to define a method able to generate results that match as closely as possible those produced by human experts. Support for such empirical stance is given by some initial evidence in the literature, indicating that some degree of convergence exists when multiple experts are asked to identify the 'important' concepts in an ontology [6].

Consistently with the stated empirical grounding of our work, we consider both criteria drawn from cognitive science as well as others based on the topological structure of the ontology. Specifically, in the initial version of our method we used both the notion of *natural categories* [7], which aims to identify concepts that are information-rich in a psycho-linguistic sense, and the notion of *density*, which highlights concepts which are information-rich in an ontological sense. In addition we also used a *coverage* criterion, to ensure that no important part of the ontology is ignored in the resulting selection. In what follows we define these criteria more precisely and present the first implementation of these ideas.

2.1 Natural Categories

Let's consider as an example the AKT Reference Ontology (AKT-RO)[3], which has been extensively analysed in a number of applications – e.g., see [5]. This ontology has been defined primarily to characterise computer science departments in academia, and would be briefly summarized by its main designer (who happens to be also one of the authors of this paper) by stating that it provides concepts to describe projects, categories of staff and students, organizations, events (in particular, academic events), technologies, publications, etc. Now, if we look at the analysis presented in [5], we can see that it indicates that, out of about 70K queries which had been posted to the AKT-RO, all but twelve focused on only four classes: Technology, Organization, Research-Area and Person. An interesting feature that links these four classes to the informal summary of the AKT-RO given by its designer is that both selections of concepts appear to be pitched at a level of abstraction akin to what Eleanor Rosch termed *natural categories* [7]. Specifically, in her seminal work, Rosch showed that people characterise the world primarily in terms of *basic objects*, such as chair or car, rather than more abstract concepts, such as furniture or vehicle, or more specific ones, such as sportscar or kitchen chair. Hence, an initial hypothesis underlying our approach was that this notion of natural categories could provide a useful basis to identify good descriptors of an ontology[4].

---

[3] http://www.aktors.org/publications/ontology/

[4] It is important to emphasise that we are by no means the first researchers to highlight the value of natural categories in identifying good descriptors of an ontology. In particular, the advantages of a *middle-out* approach to ontology design, where basic concepts are identified

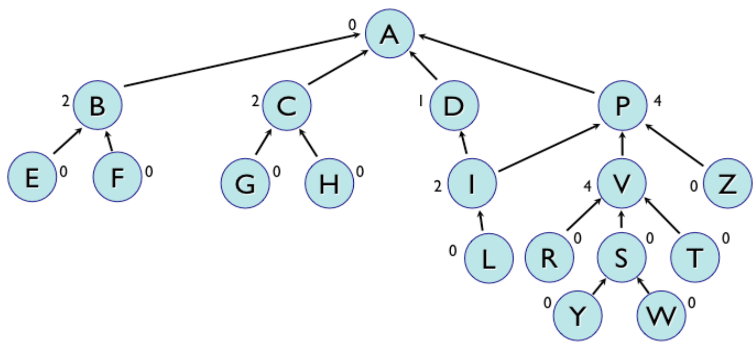

Fig. 1. Basic levels of nodes in a taxonomy – please note that measures are not normalised.

Unfortunately, to our knowledge there is no available repository of natural categories and for this reason we had to approximate this notion by devising mechanisms which operationalize it for our scenario. Specifically we have devised two measures, which we use to try and identify concepts that may play the role of 'natural categories' in ontologies.

Name simplicity. The *name simplicity*, $NS(C) \in [0..1]$, of a concept C favours concepts that are labelled with simple names, while penalizing compounds. The rationale for this criterion is that natural categories normally have relatively simple labels, such as chair or dog. In other words, they are unlikely to be compound terms. Accordingly, the name simplicity of a concept is 1 if its label is made of only one word. It decreases following the number of compounds in the label, in accordance with the following formula: $NS(C) = 1 - c(nc\text{-}1)$, nc being the number of compounds in the label and c a constant —in our experiments, we use $c = 0.3$. For example, the name simplicity of the concept *Artist* is 1, while that of *MusicalArtist* is 0.7.

Basic level. The *Basic Level*, $BL(C)$, of a concept C is a measure between 0 and 1, which indicates how 'central' C is in the taxonomy of the ontology. It is computed by counting, for each branch of the ontology containing C, how many times C can be found in the middle of a path from the root to a leaf of the branch (see Figure 1) and then normalising the value.

Given these two measures, there are two steps needed to decide the set of concepts corresponding to natural categories in a given ontology. First, the basic level and name simplicity scores are used to generate a set of candidate concepts, by choosing the ones for which $w_{BL}*BL(C) + w_{NS}*NS(C)$ is greater than a given threshold T_{nc} —in our experiments, we used $T_{nc} = 0.5$, $w_{BL} = 0.8$, $w_{NS} = 0.2$. Then, this set of candidates is filtered, by giving priority to the concepts which are neither roots or leaves of the branch, and also by assuming that only one natural category exists on a given branch of the hierarchy. If a branch contains more than one candidate concept, the one which maximizes $w_{BL}*BL(C) + w_{NS}*NS(C)$ is chosen. The output is a set of concepts, $NC(O)$,

first and used to drive the ontology development process, have long been recognized in ontology engineering [8].

which are considered as corresponding to natural categories in the context of the ontology O.

As shown by the above definitions, while natural categories in Rosch have a universal connotation, our operationalization takes into account the design of the ontology and therefore somewhat contextualises this notion with respect to the granularity of the ontology.

2.2 Topology-based criteria: density and coverage

While natural categories provide a criterion to decide what type of concepts ought to be part of an ontology summary, such a criterion is not sufficient on its own as a basis for an algorithm. We also need structuring criteria, which take into account the overall organization of the ontology. These criteria are meant to ensure that the chosen concepts embed enough information and that no important part of the ontology is left out in the 'summary'. To this purpose we also use two criteria defined on the basis of the structure of an ontology, *density* and *coverage*.

2.2.1 Density

The $density(C) \in [0..1]$ of a concept C is a measure of how richly described the concept is in the ontology and is computed on the basis of its number of direct sub-concepts, properties and instances. When computing the overall density of a concept, we use two sub-measures, *global* and *local* density. The former measures density in relation to the entire ontology, the latter only considers the neighborhood of a concept.

The global density, $globalDensity(C) \in [0..1]$, of a concept C is computed by a simple, weighted aggregation on the number of direct sub-concepts, properties and instances of C:

$$globalDensity(C, O) = \frac{aGlobalDensity(C)}{max(\{\forall N_i \in O \rightarrow aGlobalDensity(N_i)\})}$$

$$aGlobalDensity(C) = n.SubClasses(C) * w_S + n.Properties(C) * w_P + n.Instances(C) * w_I$$

In our experiments, we used $w_S = 0.8$, $w_P = 0.1$, $w_I = 0.1$.

The local density, $localDensity(C) \in [0..1]$, of a concept C refers to a density value which is relative to those of the surrounding concepts. The rationale for this measure is that, even within the same ontology, the richness of the description of concepts can vary dramatically: some areas of an ontology may contain many dense concepts, which will all be picked-up by the global density measure, while some other areas may only contain shallow concepts. For instance, the 'triangle' concept in Figure 2 is locally dense, but has a low global density, at least compared to some of the other concepts in the ontology. Hence, the local density criterion favours the densest concept in a local area, for being potentially the most important for this particular part of the ontology. It is computed using the formula below, where by "nearest concepts" to C, we refer to the set which includes sub- and super-concepts reachable through a path of maximum length 2 in the hierarchy from C, as well as C itself.

$$localDensity(C) = \frac{globalDensity(C)}{maxGlobalDensityNearestClasses(C)}$$

Finally, the overall density is computed by combining the local and global densities, each of these sub-measures being associated with a particular weight:

$$density(C) = globalDensity(C) * w_G + localDensity(C) * w_L$$

In our experiments, we used $w_G = 0.2$, $w_L = 0.8$.

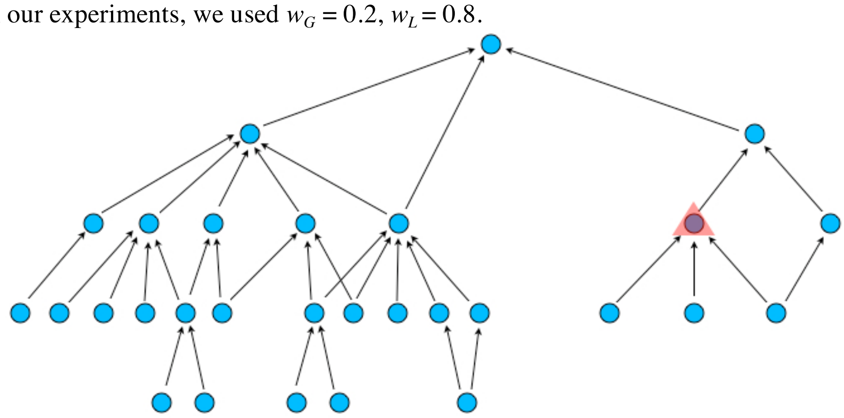

Fig. 2. Example of a locally dense concept.

2.2.2 Coverage

The coverage criterion states that the set of key concepts identified by our algorithm should maximise the coverage of the ontology with respect to its is-a hierarchy. More precisely, if $C = \{C_1, \ldots, C_n\}$ is the set of concepts returned by the algorithm and D_i is a concept in the ontology, there should be a $C_k \in C$ such that either $D_i \subseteq C_k$ or $C_k \subseteq D_i$ holds. The rationale for this criterion is that not only we want the right type of concepts to be returned by our method, but also the right spread of concepts must be achieved, to provide the best possible illustration of the ontology.

Let *Covered(C)* be the set of concepts covered by a concept *C*, i.e., *Covered(C) = C* ∪ *allSubClasses(C)* ∪ *allSuperClasses(C)*. We define *Coverage(S)* as the measure of the level of coverage of a set of concepts *S* in a given ontology. Specifically, *Coverage(*$\{C_1,\ldots,C_n\}$*)* is computed using the following formula (with $|O|$ being the size of the ontology *O* given as the number of concepts included in *O*):

$$Coverage(\{C_1, \ldots, C_n\}) = \frac{|\{Covered(C_1) \cup \ldots \cup Covered(C_n)\}|}{|O|}$$

Another useful measure related to coverage indicates how balanced a set of concepts is, i.e., the degree to which each concept contributes to the overall coverage of the set. This measure, called *Balance(S)*, where *S* is a set of concepts, is equal to the standard deviation of the elements in *S*, computed with respect to the cardinality of *Covered*(C_k), for each $C_k \in S$.

The algorithm presented in the next section requires a procedure *able to complete a set of concepts according to coverage*. That is, considering a set S of concepts of size $m<n$, we want to complete this set with additional concepts such that the resulting set is of size n, while maximizing coverage. This is realized by first computing the set S' of all the concepts not covered by S, and then generating all the possible sets, with cardinality equal to n, obtained by merging S with concepts in S'.

2.3 Key Concepts Extraction: First Version

Our algorithm takes as input an ontology, O and an integer n, with $n \leq |O|$, and returns as output n concepts in O, which best summarize it. Below we describe the algorithm in detail:

1. Using the procedure described in section 2.1, compute the set *NC(O)* of natural categories in O.
2. If the size m of *NC(O)* is
   - equal to n, then return *NC(O)* and stop.
   - greater than n, then generate the set *CandidateSets* of all the possible subsets of *NC(O)* of size n.
   - smaller than n, then generate the set *CandidateSets* of all the completed sets of concepts from *NC(O),* according to the procedure described in section 2.2.2.
3. Select the set of key concepts to return in *CandidateSets* by applying successively the following criteria, until only one candidate set is left:
   a. Restrict *CandidateSets* to the sets of concepts $S \in$ *CandidateSets*, which maximise *Coverage(S)*
   b. Restrict *CandidateSets* to the sets of concepts $S \in$ *CandidateSets*, which minimise *Balance(S)*
   c. Restrict *CandidateSets* to the sets of concepts $S \in$ *CandidateSets*, which maximise the average of $w_{BL}*BL(C_k) + w_{NS}*NS(C_k)$, where $C_k \in S$
   d. Restrict *CandidateSets* to the sets of concepts $S \in$ *CandidateSets*, which maximise the average of $density(C_k)$, where $C_k \in S$
   e. Randomly choose one set S in *CandidateSets* and return it.

Essentially, this algorithm returns a set of size n of concepts from O, which is computed by selecting concepts that appear to be 'natural categories', then taking into account how this set of concepts covers the ontology, and finally using the density of the concepts to discriminate between possible alternatives.

2.4 Evaluation of the first version of the algorithm

In order to evaluate the level of similarity between the output produced by our method and human experts, we performed an evaluation using four different ontologies: *biosphere*[5], *music*[6], *financial*[7], *aktors portal*[8].

[5] http://sweet.jpl.nasa.gov/ontology/biosphere.owl
[6] http://pingthesemanticweb.com/ontology/mo/musicontology.rdfs

We asked eight people with good experience in ontology engineering to select up to 20 concepts they considered the most representative for summarizing the contents of the ontologies. We also told them that if possible they should try and achieve a good coverage of the various parts of the ontology, rather than simply selecting all concepts from one particular branch in a taxonomy and ignore the others. In other words, we explained to them that achieving a good coverage was a desirable feature, but of course we did not give any formal guidance on how to apply this criterion, nor we mentioned the other criteria used by our approach.

Table 1. The concepts shared by more than half of the experts.

Ontology	Number of concepts in *O*	Concepts shared by the experts
biosphere	87	Animal, Bird, Fungi, Insect, Mammal, MarineAnimal, Microbiota, Plant, Reptile, Vegetation
music	91	Event, Genre, Instrument, Medium, MusicArtist, MusicGroup, MusicalExpression, Record, Sound
financial	188	Bank, Bond, Broker, Capital, Contract, Dealer, Financial_Market, Order, Stock
aktors portal	247	Computing-Technology, Geopolitical-Entity, Event, Organization, Person, Publication, Publication-Reference, Software-Technology

Table 2. Average proportion of the concepts in Table 1 selected by each expert.

Ontology	mean agreement among experts
biosphere	73.75%
music	76,39%
financial	75%
aktors portal	73,61%

Table 1 shows the concepts that were chosen by at least 50% of the experts, while Table 2 measures the level of agreement on the concepts shown in Table 1. Hence, the tables show that a consensus emerged on a number of concepts in each ontology and, for these concepts, the level of agreement among experts was good, with a mean value of *74.68%*. Indeed, it is important to emphasise that the ontologies used in our

---

7 http://www.larflast.bas.bg/ontology
8 http://www.aktors.org/ontology/portal

study are significantly larger than those used in [6], hence our experiments show that not just in small ontologies but also in medium sized ones, a degree of consensus emerges when experts are asked to identify key concepts.

Unfortunately the results for our method were disappointing. As shown in Table 3, our method only exhibits an average 42.56% level of agreement with the experts, much lower than the measure of inter-expert agreement shown above.

Table 3. Correlation between the first version of our method and the experts.

Ontology	Common choices between the testers and the algorithm	%
biosphere	Animal, MarineAnimal, Plant	30
music	Event, Genre, Instrument, MusicalExpression	44,44
financial	Broker, Dealer, Order	33,33
aktors portal	Computing-Technology, Event, Organization, Person, Publication-Reference	62,5

3. Revised Approach

3.1 What went wrong? How experts select key concepts

The analysis of the results we obtained from the experts shows that while people may employ the three criteria used by our algorithm, their application is different from the way the algorithm combines them. Our subjects did not apply coverage as strictly as our algorithm and moreover they seemed to use density ahead of natural categories. In addition, our approximation of the notion of natural category, with its emphasis on centrality and name simplicity, did not work well. Many concepts which are not natural categories may have a very simple label and, given that different ontologies have different degrees of structure and depth, centrality turned out not to be crucial, especially when it did not correlate with density. In other words, we did not find any evidence that contextualizing the notion of natural categories to the granularity of a specific ontology correlates with expert choices. Let's clarify this point with an example. Figure 3 shows some of the subclasses of the class *Animal* in the *biosphere* ontology. These have all very simple labels and have no children. However, several experts selected *Bird* and *Insect* as key concepts, even though none of the criteria we use is able to select them ahead of their siblings: they are neither dense nor central and their labels is not lexically simpler than any of the other subclasses *of Animal*.

To deal with these cases we introduced a new criterion, called *popularity*, to try and identify concepts that are particularly common, such as *Bird* and *Insect*. The advantage of this approach is that it allows us both to pick many natural categories (such as *Bird* and *Insect*) and also to identify *best exemplars* of a concept, in those cases in which we are not dealing with natural categories. Operationally, we measure the popularity of a concept, *C*, as the number of results returned by querying Yahoo with the name of *C* as keyword. Compound names are transformed to a sequence of lower case keywords separated by a space. For instance, *Marine-Animal*, *MarineAnimal*, *marineAnimal*, *marine_animal* are all transformed in "*marine animal*".

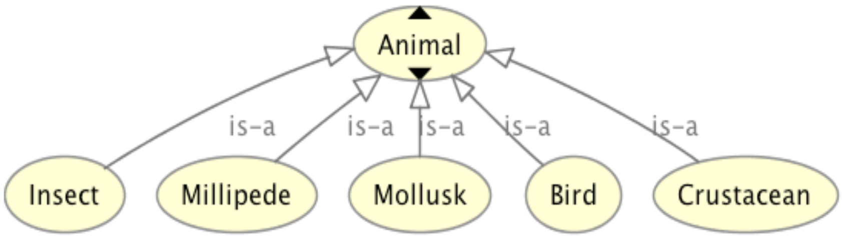

Fig. 3. Subclasses of class *Animal* in the *biosphere* ontology.

3.2 Revising the algorithm

On the basis of the considerations discussed in the previous section, we revised our method and implemented and tested two new versions of the algorithm for key concept extraction, which include the popularity criterion as well as those used in the first version of our method. For the sake of conciseness, in what follows we will focus on the third and final version of our system, which is the one exhibiting the best overall performance –i.e., the highest degree of correlation with the choices of the experts.

3.2.1 New concepts and formulas

In order to understand the new version of the algorithm, we need to introduce a number of new concepts and formulas. First of all, we want to improve the way we compute local densities, to obtain a more continuous spread of values. This is achieved by means of the following formula:

$$localDensity(C) = \frac{globalDensity(C)}{max(\{\forall N_i \in nearest_k(C) \rightarrow weightedGD(C, N_i)\})} + w_{GDL} * globalDensity(C)$$

$$weightedGD(C, N) = (1 - (ratio_D * distance(C, N))) * globalDensity(N)$$

The function $nearest_k(C)$ returns the class *C* and its sub- and super-classes, which are reachable through a path of maximum length *k* in the hierarchy. In our experiments we used $k = 2$, $ratio_D = 0.1$ and $w_{GDL} = 0.5$.

The function *weightedGD(C,N)* is used to ensure a more continuous distribution of the local density values, compared to the definition given in section 2.2.1. To this purpose, when we calculate the maximum global density value of the set $nearest_k(C)$ we take into consideration a weighted global density value for the classes $N \in nearest_k(C)$. In a nutshell, as the distance from *N* to *C* increases, the weighted global density of *N* with respect to *C* decreases.

As in the case of density, we also want to take into consideration both the *global* and *local* popularity of a concept, and we compute these analogously to the way we derive global and local densities:

$$globalPopularity(C,O) = \frac{hits(C)}{max(\{\forall N_i \in O \rightarrow hits(N_i)\})}$$

$$localPopularity(C) = \frac{globalPopularity(C)}{max(\{\forall N_i \in nearest_k(C) \rightarrow weightedGP(C,N_i)\})} + w_{GPL} * globalPopularity(C)$$

$$weightedGP(C,N) = (1 - (ratio_P * distance(C,N))) * globalPopularity(N)$$

The function *hits(C)* returns the number of hits that we obtain querying Yahoo with the name of *C* as keyword. In our experiments we used $k = 1$, $ratio_P = 0.1$ and $w_{GPL} = 0.5$.

The new version of the algorithm is based on the calculation, for each class *C* of an ontology *O,* of its local and global density, local and global popularity and its natural category value, *NCValue*, which is the normalized value of $w_{BL}*BL(C) + w_{NS}*NS(C)$, as described in section 2.1. All these measures are aggregated in a new overall value associated with a concept, called *score*, which corresponds to a weighted sum of all the above measures, as shown by the following formulas:

$$score(C) = D(C) + P(C) + NCValue(C)$$

$$D(C) = w_{LD} * localDensity(C) + w_{GD} * globalDensity(C)$$

$$P(C) = w_{LP} * localPopularity(C) + w_{GP} * globalPopularity(C)$$

$$NCValue(C) = w_{BL} * BL(C) + w_{NS} * NS(C)$$

In our experiments we used $w_{LD} = 0.32$, $w_{GD} = 0.08$, $w_{LP} = 0.1$, $w_{GP} = 0.2$, $w_{BL} = 0.66$, $w_{NS} = 0.33$.

We also extended the coverage criterion with a new function called *contribution,* which aims to measure the actual 'contribution' of a class C_i to the coverage of a set of classes $\{C_1,\ldots, C_i, \ldots, C_n\}$ in *O*, by counting the classes of *O* covered only by C_i in this set. This value is computed as follows:

$$contribution(C_i, \{C_1,\ldots, C_i,\ldots, C_n\}) = \left| Covered(C_i) - \cup_{1 \le k \le n \wedge k \ne i} Covered(C_k) \right|$$

Finally, we define the *optimal coverage* for an ontology *O* as a set $S = \{C_1,\ldots, C_n\}$, where $Coverage(S) = 1$, and each $C_i \in S$ provides the same *contribution* with respect to *S* as the other concepts in *S*.

3.2.2 Specification of the revised algorithm

As in the first version, our revised algorithm takes as input an ontology *O* and an integer *n,* with $n \le |O|$, and returns as output *n* classes in *O*, which best summarize it. In our experiments we used $n = 20$.

Table 4. Correlation between the final version of our algorithms and the experts. Concepts in *italic* in the second column are the ones also picked by more than half of the experts.

Ontology	Algorithm choices	% matches with experts' choices
biosphere	*Animal*, Bacteria, *Bird*, Crown, Fish, *Fungi*, FungyTaxonomy, Human, Litter, LivingThing, *Mammal*, *MarineAnimal*, Marine-Plant, *Microbiota*, MicrobiotaTaxonomy, Mold, Mushroom, *Plant*, *Vegetation*, Yeast	80
music	Agent, CorporateBody, Document, *Event*, Expression, *Genre*, Group, *Instrument*, Item, *Medium*, *MusicalExpression*, Musi-calManifestation, MusicalWork, OriginMap, Person, *Record*, Show, Signal, TimeLine, Work	66.67
financial	Agent, *Bond*, *Capital*, Card, Cost, *Dealer*, Financial_Asset, Fi-nancial_Instrument, *Financial_Market*, Money, *Order*, Organization, Payment, Price, Quality, Security, *Stock*, Supplier, Transaction, Value	66.67
aktors portal	Educational-Organization-Unit, Employee, *Event*, Information-Bearing-Object, Intangible-Thing, Integer, Legal-Agent, Loca-tion, Message, Month, Number, *Organization*, *Person*, *Publica-tion*, *Publication-Reference*, Set, *Software-Technology*, Tech-nology, University, Working-Person	75

Below we describe the algorithm in detail:

1. For each class C in O we compute its global and local density, global and local popularity and the natural category value.
2. For each class C in O we compute *score(C)*, as described in section 3.2.1.
3. Given a number $k \leq n$ (in our experiments $k = 15$), let S be the set of k classes in O with the best *score* and let T be the set of n-k classes in $\{O - S\}$ with the best *score*. If T is empty, we return S and we stop.
4. Otherwise, let c be the average of all the values obtained by invoking the function *contribution*(C_i, $\{S \cup T\}$), for each $C_i \in \{S \cup T\}$. And let a be the average of all the values obtained by invoking the function *overallScore*(C_i, $\{S \cup T\}$), again for each $C_i \in \{S \cup T\}$. The function *overallScore* is defined as follows.

$$overallScore(C_i,\{C_1,\ldots,C_i,\ldots,C_n\}) = w_{CO} * \frac{contribution(C_i,\{C_1,\ldots,C_i,\ldots,C_n\})}{maxContribution(\{C_1,\ldots,C_i,\ldots,C_n\})} + w_{CR} * score(C_i)$$

   In our experiments we have used $w_{CO} = 0.6$ and $w_{CR} = 0.4$.

5. Let W be the class in T with the worst *overallScore*(W, $\{S \cup T\}$) of all the classes in $\{S \cup T\}$, and let R be the set $\{\{S \cup T\} - \{W\}\}$. If there is a class $B \in \{O - \{S \cup T\}\}$, such that
   (a) the average a' of all the values obtained by invoking *overallScore*(C, $\{R \cup \{B\}\}$), computed for each $C \in \{R \cup \{B\}\}$, is greater than a,

(b) the average *c'* of all the values obtained by invoking *contribution(C, {R ∪ {B}})*, computed for each *C* ∈ *{R ∪ {B}}*, is greater than or equal to *c*,

we swap *W* with *B* in *{S ∪ T}* and we go back to step 4. Otherwise we return *{S ∪ T}* and we stop.

3.3 Evaluation of the revised version of the method

The tests performed with the new version of the algorithm produced much better results than the previous version. In particular, as shown in table 4 the average measure of agreement between our algorithm and the human experts is now *72.08%*, only 1.5 points lesser than the inter-expert agreement (*74.68%*). In practice the final version of our method, at least on the current benchmark, is indistinguishable in its output from human experts.

4. Related Work

As already mentioned, a few papers have addressed the topic of ontology summarization. In particular, in [6] the authors describe a family of algorithms to select the *salient RDF sentences* from a RDF graph. These algorithms work primarily on the basis of the topological structure of the graph. The paper shows that while there is a relatively low correlation between experts at the sentence level, there is a much better degree of agreement with respect to vocabulary overlap. In addition, they also show that the results produced by their method exhibit a good degree of correlation with the experts. However, the ontologies used in their case studies are much smaller than the ones used here, so those results are potentially less significant than those presented here, even though no firm conclusion can be stated without trying out both approaches on a common benchmark. The work described in [5] focus on *winnowing* an ontology – i.e., reducing the size of an ontology to facilitate its reuse. Hence, in this work the focus is on a different type of summarization, which aims to make the ontology more easily reusable, rather than facilitating ontology understanding in a context in which the user wishes to quickly get a snapshot of what an ontology is about. The same consideration applies to work on *ontology customization* [9], which provides mechanisms to enable particular views over an ontology. While this work can be seen as a particular kind of ontology summarization, it differs from our work both with respect to the output of these techniques (a particular *cut* over an ontology) and also because it expects the user to specify which part of an ontology she is interested in.

5. Discussion

While the generation of automatic methods, able to extract ontology summaries in a way which correlates with human experts, is an interesting research issue in itself, the work described here also provides a potentially useful basis for a number of novel contributions to ontology engineering and semantic web research. In section 1, we have already pointed out that a key motivation for this work was to facilitate the process of ontology understanding for users of the Watson ontology search engine and the Watson Plug-in. In particular, by providing quick snapshots of an ontology as part of the results returned by Watson, we hypothesise that it will be easier for users to quickly home in on the ontology most relevant to her needs. We also plan to use this work as the basis for a novel visualization algorithm, to complement and to address the weaknesses of the traditional taxonomic-centric support for navigating ontologies, which is provided by current ontology engineering editors. As discussed in [10], classic hierarchical views of ontologies are not very helpful for supporting tasks related to understanding the general structure of an ontology. In particular, consistently with the experiments carried out here, the concepts that experts select to describe an ontology tend to be on different branches of the hierarchy at various levels of depth. Hence, they cannot be easily identified with standard top-down taxonomy browsers.

Initial presentations of this work to a number of audiences have also elicited interesting suggestions for applying the work described here to a number of ontology-centric scenarios. In particular, colleagues have suggested the use of our summarization technique in scenarios where an ontology is used to support automatic data classification, but it is too expensive to try and classify large quantities of data against a large number of classes. In these scenarios, our method could be used to identify the most useful concepts in an ontology, so that these can be tried first. Similar ideas have been suggested by colleagues working on ontology matching and evolution, where the ability to prioritize which concepts the system ought to focus on could also be useful. Analogously, key concept selection could also be used as the basis for a new family of ontology modularization algorithms. For example, modules could be built around each key concept, so that the resulting partitioning of the ontology would identify 'key areas' of the ontology, consistently with the criteria presented in this paper. Finally, we also intend to use this method in the context of the work on 'cautious knowledge sharing', which we are carrying out in the OpenKnowledge project (http://www.openk.org). This work is concerned with scenarios where the content of an ontology is proprietary or otherwise restricted, and cannot be made publicly available. In these scenarios, automatic ontology summaries can be useful as a way to advertise an ontology while disclosing as little content as possible.

6. Conclusions

In this paper we have introduced a user-independent approach to identifying automatically the key concepts in an ontology. The approach integrates both topological measures, such as density (both global and local) and coverage, as well as statistical lexical measures (popularity), and cognitive criteria (natural categories). The approach has been validated empirically, by showing that the revised version of our implementation shows an excellent degree of correlation with human experts. However, we should stress that these results, although promising, are still preliminary. A more extensive evaluation study will be needed, to determine more conclusively both the extent to which experts are able to agree on what are the best concepts to describe an ontology and also the extent to which this approach can emulate expert concept selection in a variety of domains. It will also be interesting to extend the algorithm, so to be able to add also 'key properties' and even 'key individuals' to the ontology summaries. In particular, adding key properties introduces interesting issues, as some degree of coherence needs to be ensured between the set of concepts and the set of properties identified by the algorithm. Hence, a possible strategy could be to focus on concepts first, using the approach presented in this paper, and then extend such selection by identifying the most important properties associated with the selected concepts, rather than with the ontology as a whole.

As already mentioned, we also intend to apply these ideas to a number of ontology engineering tasks, e.g., to explore new approaches to ontology visualization and navigation, ontology evolution, and ontology modularization. Finally, we plan to make our system available as a resource for the ontology engineering and semantic web communities, by exposing it as a web application.

Acknowledgments. This work has been partially funded by the OpenKnowledge IST-FP6-027253 and NeOn IST- FP6-027595 projects. The authors would like to thank an anonymous referee for his numerous insightful suggestions.

References

1. d'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., and Motta, E. (2007). Characterizing Knowledge on the Semantic Web with Watson. 5th International EON Workshop, co-located with the 6th International Semantic Web Conference (ISWC'07), Busan, Korea, September 2007.
2. Stefik, M. (1986). The next knowledge medium. AI Magazine 7(1), pp. 34-46.
3. d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., and Guidi, D. (2008). Towards a New Generation of Semantic Web Applications, IEEE Intelligent Systems, 23, 3, pp. 20-28
4. Dzbor, M., Motta, E., Buil Aranda, C., Gomez, J.M., Goerlitz, O., and Lewen, H. (2006). Developing ontologies in OWL: An observational study. Workshop on OWL: Experiences & Directions, November 2006, Georgia, US.
5. Alani, H., Harris, S., and O'Neil, B. Winnowing Ontologies based on Application Use, Proceedings of 3rd European Semantic Web Conference (ESWC), Budva, Montenegro, 11-14 June 2006.
6. Xiang, Z., Cheng, G., and Qu, Y. Ontology Summarization Based on RDF Sentence Graph. 16th International World Wide Web Conference (WWW2007), Banff, Alberta, Canada, May 8-12 2007.
7. Rosch, E. Principles of Categorization, Cognition and Categorization, Lawrence Erlbaum, Hillsdale, New Jersey, 1978.
8. Uschold M. and Gruninger M. (1996). Ontologies: Principles, Methods and Applications. Knowledge Engineering Review, 11(2), pp.93-136.
9. Bercovici, N., Gröner, G., Schenk, S., Kubias, A., and Dzbor, M. (2008). Ontology customization and module creation: query-based customization operators and model. NeOn Project Deliverable D4.2.2, Feb. 2008.
10. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E. (2007). Ontology Visualization Methods - A Survey. ACM Computing Surveys, 39(4), 2007.