KEY CONCEPTS EXTENDED BY VECTOR DESCRIPTIONS TO INTERPRET THE MEANING OF ONTOLOGIES

Jozef VRANA, Marián MACH
Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, tel.: +421 55 602 4214, e-mail: jozef.vrana@tuke.sk

ABSTRACT

The Semantic Web has reached the point where basic research is supposed to be replaced (at least partially) by research aiming at practical aspects of using the Semantic Web. Ontologies, as the essential technology in this area, have been under the spot light which produced some results, e.g. the semantic web search engines like Watson or Swoogle. These search engines help in finding and locating semantic information on the Web. However, they do not support users to quickly understand what an ontology is about, what information it contains. We argue in this paper that vector description, as a snapshot of data comprised in ontologies and therefore representing a vector-based gold standard of a domain, may help while trying to understand a particular ontology. In other words, instead of depicting a whole ontology all together, we prefer reduction of information given and therefore avoid users being overwhelmed and thus help with orienting in a wide offering of ontologies on the Web.

Keywords: *ontology, ontology evaluation, ontology visualization, ontology key concepts, ranking ontology objects*

1. INTRODUCTION

Current work in Knowledge Management, the Semantic Web, and a variety of Semantic Web Services depends on ontologies serving as a backbone to application development [1]. In our work we argue for the need to develop an easy way to quickly obtain a general impression of what a particular ontology is about. It is believed that metrics are essential to achieve significant progress on the field of development and deployment of ontologies, evaluation metrics must be available. Whereas there are several semantic web engines (e.g. Watson [2], Swoogle [3]) that can be used to retrieve ontologies, only very few methods exist to support decision making about which ontologies are suitable for certain purposes [4]. Preliminary studies [5] revealed the high level of dissatisfaction with existing tools among users what signalises that there is a demand for a simple and comprehensible tool enabling to evaluate ontologies. Users mostly commented on excessively high complexity of information given, not being able to cope with performing basic actions without any previous training [4, 5]. They often feel overwhelmed by the amount of information provided and lost when being confronted with large scale ontologies. The conclusions, we drawn, are that only limited data are required in order to determine the suitability of an ontology for a particular domain.

As a part of the tool, the development of which is being carried out, we created a way of measuring terms coherence regarding its semantics. Term interpretation leads to many problems since terms are too ambiguous, what makes them impossible to compare mutually without additional data (e.g. words *exploit* and *explore* have semantically nothing in common even though they differ in just two characters). Homonyms are yet another example of how words cannot be interpreted without context (e.g. car in the meaning *elevator car* is semantically closer to the term *building* than car as a *cable car*). In this paper we present the notion of vector description that can be used as a supplementary resource explaining the meaning of ontology terms.

The next section compares our technique against existing methods of ontology evaluation and discusses the motivation behind our research. Then it is explained the process of gathering, weighting and processing description vectors. As a part of the research, studies were carried out, results of which are presented together with conclusions at the end.

2. PROBLEM OUTLINE

Current ontology evaluation is an essential part of informaion reuse even though very little attention has been paid to it. Ontology evaluation is the problem of assessing a given ontology from the point of view of a particular criterion (usually set by the application), in order to determine which of several ontologies would best suit a particular purpose [6]. Methods for the evaluation of ontology scope may be categorized based on what sort of information from an ontology is used to determine a domain:

- hierarchy and taxonomy layer
- lexical vocabulary or data layer

Whereas the former takes advantage of relations among objects, their position in the taxonomy and the number of bonds with other objects, the latter one is mostly aimed on textual data comprised by ontology. In the presented approach both abovementioned approaches are adopted and so the lexical content is taken into consideration as well as hierarchical information.

The two categories mentioned above are the only groups relevant to our approach though there are other methods such as *context or application level* where ontologies are being evaluated by other, contextually bounded ontologies. Some groups are solely reserved for manually constructed ontologies (e.g. *Structure, architecture, design*) where ontology is expected to meet certain pre-defined design principles or criteria.

Ontology evaluation techniques usually compare the ontology to be evaluated against a golden standard and afterwards decide on a domain of it. This might sound

logical, comparing two formal resources to each other, but it assumes the existence of the golden standard ontology for any area of interest imaginable. Other approaches require participation of human experts or using the ontology in test application. Each of the above mentioned approaches has its own shortcomings though there is one that they share: both of them require the existence of human-created standards. The problem with the golden standard approach is that if the evaluation results differ from the standard, it is hard to decide whether this is because the standard is inappropriate, the methodology is flawed or there is a real distinction between the knowledge present in the ontology and the golden standard.

One way of approaching the problem might be to decompose it into its constituent parts. An ontology is composed of concepts and relations, some of which are explicitly defined while the others follow from a set of axioms. Disintegration of an ontology into the set of concepts yields us an annotation of the ontology in the form of terms ordered according to their importance. Applying that approach on a set of ontologies (from the domain we are describing) further extends the number of terms acquired and creates a new golden standard (in the form of a vector) that is automatically generated and accurate concerning the domain. In comparison with the traditional golden standard approach, our method has two major advantages:

- vectors are easier to compare than complex ontologies,
- vectors are acquired automatically from an ontology corpus.

The former describes naturality of vector notion in computer science and thus enables more efficient work in contrast to complex multidimensional tree structures where each object is defined by a range of attributes while the vector captures all the possible attributes in one number as *weight of term*. The bottom line is, therefore, all the calculations are performed with numbers rather than ontology objects in all their complexity. The latter identifies an acquisition process for building vector descriptions out of an ontology corpus which consequently means more accurate results than a golden standard generated by a human.

What is suggested here is comparing objects from an ontology and vector descriptions that represent different domains and by quantifying these matches we are able to assign a suitable domain to the certain ontology.

The vector driven evaluation (as we marked our approach) is the most essential premise underlying our work and in the next section it is described in more detail.

3. BUILDING VECTOR DESCRIPTIONS

Before comparing and evaluating an ontology we must have a reference entity representing the domain. A set of terms which fit into the particular domain with various degrees (represented by weights) seems to be a fairly solid approximation of the domain. These vectors are generated automatically, using all relevant ontologies from the corpus. Thus, the vector not only captures real circumstances but also it is as precise as all ontologies relevant to the domain (included in the used ontology corpus), which makes it a perfect object of reference in terms of representation as well as the way it is being generated.

Fig. 1 represents a rough scheme of how the vector building is performed. A keyword is an input, around which the sub-trees are extracted from all relevant ontologies. Ontologies can be only described as relevant if any of their concepts contains the keyword. The concept comprising the keyword (either in local name or label) is then called the original concept.

Selected entities along with their types and states regarding an original concept are therefore used for weights computation. The result of *Term extraction/Weights computation* is a vector that has not been normalized and cleaned. After cleaning/normalization the whole process is finished.

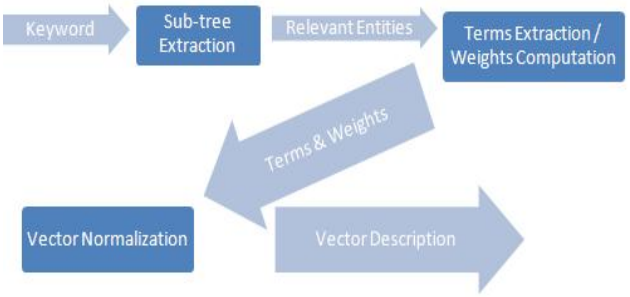

Fig. 1 Scheme of generating vector description

In other words: the domain of interest is in our case represented by a keyword. The keyword was obtained as input from the user and it is used to find other terms from the domain. These terms interpret the keyword by a vector description, containing other terms that are related to the given keyword and therefore further specify the keyword (e.g. for domain specific keyword *Academic Employee* the specifying terms may be something like: *Lecturer, Researcher, PhD Student, University, Person, Education, etc.)*.

The search engines (e.g., [1, 2]) are keyword based and so they can find ontologies where a specified keyword has emerged in the names of classes, instances and other concepts (i.e. *properties, labels* ... which are common textual data comprised in ontologies). From our perspective, the original concept is the starting point for building the vector corresponding to the keyword. When the original concept is found in an ontology, we tend to explore the nearest neighbors (sub/super related concepts) following e.g., *isA* links or any other bound to/from the original concept.

Afterwards, the obtained sub graph of an ontology is being processed and decomposed into terms. In order not to lose data carried by links, these are reflected into initial weights *IW*.

As depicted on Fig. 2, surrounding concepts extend the definition of the original concept found in the ontology. At first, initial weights are established for the original concept *Academic Employee* (IW_O stands for):

- **Class:** $IW_O = 10 + G$
- **Individual:** $IW_O = G$
- **Property:** $IW_O = 1$;

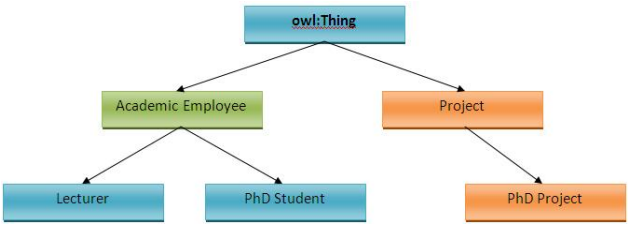

Fig. 2 Picture shows an ontology where the concept *Academic Employee* matches the keyword and so it is the centre of a sub-tree. The sub-tree (left branch of the tree) is a set of concepts taken into consideration while building a vector description for the keyword.

IW for contextually-bounded objects can be derived from the initial weight of the original object and the number of words in the label/local name of object (when no label is defined then local name is taken) as follows:

- $IW = (IW_O + G) / nbW$

for super objects (*owl:Thing* in our case is the only super object),

- $IW = IW_O / nbW$

for sub objects (*Lecturer* and *PhD Student*).

```
<owl:Ontology rdf:ID="myontology"/>
  <owl:Class rdf:ID="PhD_Project">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Project"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Lecturer">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="AcademicEmployee"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#AcademicEmployee">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Employee from the Academic sphere</rdfs:label>
  </owl:Class>
  <owl:Class rdf:ID="PhDStudent">
    <rdfs:subClassOf rdf:resource="#AcademicEmployee"/>
    <rdfs:seeAlso>
      <PhD_Project rdf:ID="Ontologies_evaluation"/>
    </rdfs:seeAlso>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="hasMember">
    <rdfs:domain rdf:resource="#PhD_Project"/>
    <rdfs:range rdf:resource="#AcademicEmployee"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="supervisedBy">
    <rdfs:range rdf:resource="#Lecturer"/>
    <rdfs:domain rdf:resource="#PhDStudent"/>
  </owl:ObjectProperty>
  <PhDStudent rdf:ID="TE001234">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Jozef Vrana</rdfs:label>
  </PhDStudent>
```

Fig. 3 Source code of the same ontology as on Fig. 2

The label of each object typically consists of one or more words (e.g. "*Academic Employee*"). The number of words in label for each concept is *nbW*. The constant G that emerges in almost every formula is a value that reflects the users' requirements in terms of generality. In other words: a user can prefer more general objects to less general objects. Thus, a user is given the opportunity to express the preference by a number on the scale from 1 to 10 (1=least general; 10=very general). The G parameter therefore does not have any effect on the cardinality of a vector (that always remains) but just changes weights within the vector. In our particular case (assuming that only ontology was found and G=1) the vector description would look like: **Keyword**: *Academic Employee*; **Vector:** ("*Academic Employee*", 11); ("*Thing*", 12); ("*Lecturer*", 11); ("*PhD Student*", 5.5).

The system does not distinguish between a keyword found as a token of larger text or an exact match and so it does no difference in final weights. The described calculation is performed on each ontology, where the keyword was found. The initial weight for a particular term reflects its importance regarding certain ontology, in case we want to extend it on ontology corpus, weights must be aggregated for each occurrence of the term in all ontologies in corpus (*finalW*).

The specification of ontology languages allows us to create an annotation of any object either in label or comment. Whereas the label tends to be a short phrase referring to the name of the object, the comment is more complex and longer text describing the object. We only look at the labels and do not consider comments. Some ontologies have blank labels and their developers encrypt the label in the local name of the object. Even though this is not the way it ought to be done, our system deals with it by taking the local name as the second choice.

Despite the nature of labels the stopwords elimination is necessary (e.g., label may be "*person with cat and dog*" and in this particular case two words must be removed: *"with"*, *"and"* as they have no information value). The principle of eliminating stopwords is well established in the area of information retrieval [7] and it is used as well by search engines that eliminate them from queries as these words have little to do with information being sought by searchers. When done early in the indexing process, the elimination of stopwords can make further processing of the candidate index terms more efficient and reduce the storage space [8]. Using this technique in text mining, stopwords are language dependent, domain specific and may reduce recall in some cases. Despite these drawbacks there is a sense in using stopwords since it helps us to genuinely increase precision. The only alternation was the addition of word *"thing"* to the stop list because it represents an abstract entity on the top level and as such carries no information whatsoever.

At this stage the initial weights are being aggregated in case one term is present more than once in the vector (a term may occur in several ontologies and so this number of occurrences is reflected within the vector) according to the formula:

$$finalW = \sum_{i=1}^{n} IW_i \quad (1)$$

finalW is referring to the weight acquired by the aggregation whereas n is the number of occurrences of the particular term in the vector. Finally, the weights are being normalized (formula 2) at the end of the process and sorted decreasingly.

$$\overrightarrow{w_i} = \frac{w_i}{\max_{u \in W} w_u} \quad (2)$$

Term frequency has been proven useful in information retrieval mainly because of its simplicity [7]. Despite the term frequency being an enhancement of term-weighting,

its use in isolation cannot ensure acceptable retrieval performance. Specifically, when the high frequency terms are not concentrated in a few particular documents, but they are instead distributed across the whole collection, all documents tend to be retrieved - and this severely affects the search precision. Here is where *inverse document frequency* comes along to help to suppress the negative aspects of using term frequency solely.

The *inverse document frequency* is generally computed according to formula 3 where *N* refers to total number of documents; *n* is number of documents containing specified term. As you will see in the forthcoming section, the idf measure was adapted to our conditions.

$$idf = \log_{10} \frac{N}{n} \tag{3}$$

The approach we have depicted in this section was applied before in text mining and it is based on the assumption that the distance of words in a plain text also reflects their semantic distance [9]. The essential premise remains though as it was applied on ontologies, the technique was tailored to this kind of information repository. In terms of accuracy results, it is believed that explicit links in ontologies are more reliable information resource compared to semantic relations based on statistical analysis calculations.

4. IMPLEMENTATION LIMITATION

Extracting keywords from a domain has been a challenging task for researchers from different fields, e.g. statistical analysis, artificial intelligence or natural language processing. Despite this interest, it has not been sufficiently solved as all techniques struggle to process text and transform it into knowledge [10]. We aimed our attention onto knowledge already and explicitly captured in the form of an ontology repository rather than ambiguous textual data.

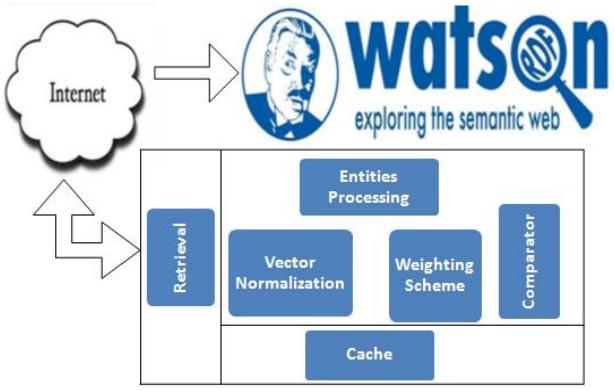

Fig. 4 Block scheme of described system

The search engine Watson [2], which has been developed at Knowledge Media Institute, was employed in the role of the ontology repository. This system as such is extremely powerful and so has an ability of acquiring high quality results. Even though Watson can cope with user requests within a short time, transferring massive amounts of data via the Internet is simply too demanding to deal with. The better solution appears to be placing the vector computation on the side of the Watson server which would provide two advantages:

- significant reduction of client-server communication which is currently the cause of delays and in consequence of that the system is not as fast as it could be;
- always updated vectors: in an effort to reduce computation time, the system caches vectors and reuses them in the future. This obviously leads to not updating the vectors. As a result, the cached vectors do not cover ontologies which were indexed after the vector computation.

The proposed solution (of placing the system on the Watson server) may appear the best though we have no access to the server and also no right to change the architecture. A more realistic approach is therefore using *Cache Management System* [11].

The reason why *idf* had to be adapted to our technique is that retrieving term frequency in the rest of the corpus causes further delays that as a result practicality of the proposed approach. The rest of the corpus comprises all ontologies, which contain particular terms except the one that is being viewed. What we did though was making a use of vectors which we have access already (typically user compares ontology against several similar domains for which the description vector was generated and these are accessible) and compute the inverse document frequency out of them, considering occurrences of tested terms in other vectors while these vectors play the role of corpus. In other words: the *idf* factor varies inversely with the number of vectors n to which a term is assigned in a collection of N vectors (according the formula 3).

Moving the vector generator to the side of the Watson server would allow employing the *idf* measure exactly like it is defined in information retrieval and so we hit the wall, once again. For completing implementation of *idf* is obligatory to recall the number of documents containing the term which would considerably increase amount of data needed to transfer and also time of computation. When vector computation is a part of the Watson server, the time consuming communication will vanish.

5. ACHIEVED RESULTS

The above section of observations motivated us to perform tests with emphasis on validity of the premises (number of ontologies necessary to gain vector description; vector description can by employed as a *golden standard* of the domain) presented in the previous sections. Together with essential ideas, the implementation issues arose and so we want the tests to sort out those as well.

In spite of describing the full process of vector generation, we have not presented an actual vector yet. Fig. 5 presents a vector description. In the first line there is the keyword, for which the vector was created. Below the keyword, there are terms acquired. Terms are presented together with their weights.

Taking into account the requirement on as short time of computation as possible, we try to save the computation time wherever we can. One way of achieving that may be reducing the number of ontologies needed as input for the

vector generator what basically means reducing the corpus. An unpleasant consequence of such reduction might be decrease in accuracy of obtained vectors as these would be computed on smaller a collection of ontologies.

```
Phd Project
project-1.0;person-0.23;student-0.172;organization-0.142;
activity-0.115;doctor-0.098;document-0.096;research-0.093;
funding-0.081;publication-0.077;unit-0.068;member-0.063;
degree-0.049;leader-0.036;academic-0.027;philosophy-0.022;
research project-0.019;paper-0.016;support-0.011;
description-0.0080;role-0.0070;has project leader-0.0050;
area-0.0010;
```

Fig. 5 Vector description for the keyword *PhD Project* as represented in the cache

The test on Fig. 6 proves that the higher number of used ontologies, the higher cardinality of the vector. Along with this logical consequence, there is also a weights adjustment. Some objects co-occur more often than the others which causes preference of frequent terms to the rare ones.

```
Instrument - 10 Ontologies
instrument-1.0;artifact-0.555;obi-0.385;woodwind-0.242;musical-0.227;sport-0.195;
measuring-0.171;percussion-0.159;triangel-0.136;stringed-0.111;brass-0.1;device-0
.093;optical-0.088;musician-0.083;cooking-0.071;weapon-0.055;medical-0.053;string
-0.047;solo-0.046;rub-0.041;28instrument-0.038;accessory-0.035;main-0.031;templat
e-0.029;0000311-0.027;cello-0.025;hand-0.022;guitar-0.021;horn-0.019;jpg-0.018;cl
ock-0.013;triangolo-0.012;stove-0.01;instrument used in fencing-0.009;instrument
for archery-0.006;0400113-0.004;an instrument used for measuring altitude
particularly in an air vehicle-0.003;list-0.001;

Instrument - 176 Ontologies
instrument-1.0;commodity-0.754;artifact-0.26;obi-0.253;musical-0.219;object-0.178
;class-0.157;musician-0.155;msi-0.151;document-0.146;event-0.138;woodwind-0.128;p
rocess-0.102;measuring-0.099;sport-0.091;owl-0.086;percussion-0.079;patient-0.077
;equipment-0.066;stringed-0.065;triangel-0.064;role-0.061;accessories-0.06;400059
-0.05;www-0.047;property-0.044;parts-0.043;timepiece-0.041;plays-0.04;keyboard-0.
039;medical-0.035;piano-0.034;talkdigger-0.033;category-0.03;hardware-0.028;weapo
n-0.027;financial-0.026;independent-0.025;viola-0.024;english-0.023;accessory-0.0
21;used-0.02;semantic-0.019;cello-0.018;0400003-0.017;trumpet-0.016;collection-0.
015;audio-0.014;bass-0.013;0000311-0.012;sourceforge-0.011;clarinet-0.01;certific
ate-0.009;communication-0.008;marimba-0.007;triangolo-0.006;laryngoscope-0.005;in
strument used in fencing-0.004;deed-0.003;sight-0.002;mass spectrometry
instrument-0.001;
```

Fig. 6 Illustration of two vectors for the keyword *Instrument,* which differ in their cardinality and some weights are different as well

To sum it up, on Fig. 6 two vectors are presented values of which are ordered decreasingly. As the number of retrieved ontologies increases, some terms amplify their weights (*musician* became more important regarding *instrument*) while some term disappeared due to inability to overcome threshold value (rounding to 3 decimal places causes that all smaller weights are dropped out).

The number of used ontologies (test on Fig. 6) is a restriction that is used to confine the number of ontologies though it does not say anything about the actual number of relevant ontologies in the corpus. Thus the numbers reflect more of maximum range rather than the count of relevant ontologies in the corpus. This is why (on Fig. 6) we used 10 and 176 ontologies. There are simply only 176 ontologies in the Watson corpus containing *Instrument*.

Vectors presented on Fig. 5 and 6 are in their final state, in which all steps of the process were done, including normalization and tf-idf. These two steps have no result on the cardinality of the vector and only change significance regarding the vector (weights). The forthcoming test reveals how weights change after applying tf-idf measure.

```
lecturer
lecturer:1.0;teaches:0.516;academic:0.489;student:0.258;staff:0.247;portal:0.194;
academia:0.172;professor:0.172;room:0.172;teached:0.172;faculty:0.129;lecturer in
academia:0.129;senior lecturer in academia:0.129;taught:0.129;senior
lecturer:0.065;member:0.054;person:0.054;agent:0.032;assistant
lecturer:0.032;employee:0.032;http  www new onto org 1054569311671
lecturer:0.032;interests bml int lecturer:0.032;lapzwans:0.032;lecturing:0.032;le
hrbeauftragter:0.032;teacher:0.032;position:0.022;type:0.022;00030:0.016;sep:0.01
6;akt:0.014;ontology:0.014;owl:0.014;activity:0.008;occupation:0.008;org:0.005;

lecturer
lecturer:1.0;teaches:0.79;academic:0.749;staff:0.378;portal:0.283;academia:0.263;
professor:0.263;room:0.263;teached:0.263;student:0.258;faculty:0.197;lecturer in
academia:0.197;senior lecturer in academia:0.197;taught:0.197;senior
lecturer:0.095;member:0.083;person:0.079;agent:0.049;assistant
lecturer:0.049;employee:0.049;http  www new onto org 1054569311671
lecturer:0.049;interests bml int lecturer:0.049;lapzwans:0.049;lecturing:0.049;le
hrbeauftragter:0.049;teacher:0.049;position:0.034;type:0.034;00030:0.024;sep:0.02
4;akt:0.021;ontology:0.021;owl:0.021;activity:0.012;occupation:0.012;org:0.008;
```

Fig. 7 The former vector is build using tf solely whereas the later one contains tf-idf weights

As you may notice on Fig. 7, adding *idf* measure into the equation had a consequence in alternating the position of certain terms. It is important to bear in mind that *idf* is calculated on few vectors and so results may be affected considerably by the composition of these vectors.

Academic employee

	Instrument	Phd_Project	Student	Education	Music	Supervisor	Entertainment
10	0,0000	1,3735	0,3254	0,2309	0	0,3799	0,1176
100	0,3653	1,1952	0,3123	0,2637	0,0748	1,1952	0
200	0,3627	0,8441	0,5914	0,1938	0,1917	0,9006	0,2397
300	0,2898	0,8636	0,5932	0	0,1842	1,0024	0,2128
400	0,1979	0,8413	0,503	0,1568	0,1615	0,9876	0,1622
500	0,1652	0,9336	0,3252	0,3186	0,1573	0,9802	0,1608

Project

	Instrument	Phd_Project	Student	Education	Music	Supervisor	Entertainment
10	0,2336	1,3967	0,2732	0,0000	0,2512	0,3005	0,2512
100	0,1307	1,4899	0,2917	0,2095	0,1336	0,2535	0,0943
200	0,1596	1,5745	0,2596	0,0697	0,1536	0,5357	0,1488
300	0,1289	1,4686	0,1929	0,111	0,1241	0,2228	0,1178
400	0,1394	1,4869	0,2543	0,0556	0,1621	0,2177	0,102
500	0,0618	1,5228	0,1894	0,0569	0,0774	0,2328	0,1024

Object

	Instrument	Phd_Project	Student	Education	Music	Supervisor	Entertainment
10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
100	0,6989	0,0934	0,0258	0,6696	0,5357	0,0158	0,1709
200	0,5608	0,1419	0,1244	0,5367	0,6098	0,1427	0,4331
300	0,8603	0,1965	0,2148	0,3441	0,8531	0,1772	0,5337
400	0,8376	0,2084	0,2729	0,3891	0,8852	0,2125	0,4709
500	0,8376	0,2187	0,3243	0,3872	0,8362	0,2185	0,487

Fig. 8 Tables represent similarity between a vector (*Academic employee, Project, Object*) and description vectors that stand for a background (an environment where objects are set in). Numbers in cells are measures of similarity whereas rows refer to the number of ontologies used for computing the vectors.

In order to respond to the issues identified (such as accuracy of assessing domain; how the number of used ontologies affects this accuracy), we developed a test relying on generating vector descriptions for three domains, using different settings. The tables on Fig. 8 show three keywords (*Academic employee, Project, Object*) for which the vectors were generated using

different numbers of ontologies (*10, 100, 200 ...)*. On the other hand the columns are keywords representing different domains.

These domains are also vectors but generated regardless of the number of ontologies (above the whole ontology corpus). The results of comparing every keyword with each domain are measures of similarity (values in the tables). These reflect equality among keyword vector and all domains. The vectors are being compared by finding all equal terms and averaging their weights from both vectors. Tables were generated in order to decide on the number of ontologies necessary to obtain an appropriate vector description. Dark colours symbolize higher values what consequently implies assignment of a particular object into one or more scopes. It is crucial to point out, that the number up to 300 ontologies used, has impact on results but afterwards the changes are minor and so 300 is taken as a sufficient number of ontologies (but an optimal number can vary based on, for example, the generality of the used keyword and quality of used ontologies).

There is no clear threshold value to establish a *good match* between a keyword and a domain. All the similarities must be perceived in context of values in other columns (Fig. 8), for instance keyword *Academic employee* has closer bound with *Supervisor* than the *PhD Project.* In this sense, the domains constrain the environment in which the keyword is depicted and numbers determine its position in reference to domains.

Also what captures the eye is that the resemblance of certain vectors is fairly clear for 10 ontologies and becoming even clearer with the number of ontologies rising (if values in all columns are more less the same or differ very slightly, the problem in either in the weighting scheme or inappropriately chosen domains). This assumption apparently does not apply on *Object* (the third table on Fig. 8) which can fit in several scopes of background probably due to the nature of the term *Object* which is too general.

It is necessary to say that the numbers presented on Fig. 8 are not normalized as they are used exclusively on test purposes. Furthermore, measures of similarity do not reflect probability but the level of equality.

6. FUTURE WORK

Future motivation is therefore to build upon this technique and to develop a more comprehensible tool for ontology evaluation; the tool with no extra requirements on user; the tool that takes care about everything without the user even noticing complex computation behind the scene.

The spotlight is now redirected onto more efficient capturing of data, comprised by links among objects. There are further tests necessary with an aim of finding the best scheme of how to use relations in our system.

Presented measuring systems also lacks deeper research in terms of comparing two vectors mutually. Research area of data processing offers a range of techniques some of which are relevant to our problem.

Put aside implementation issues, the vector as such is not the most suitable representation and therefore an additional system for interpretation is mandatory. Far more appropriate is presenting pictures with vectors standing behind. This would be our motivation for the future: using this specific data derived from ontologies to introduce a new approach to ontology (vector driven) evaluation and an appropriate visualization technique.

Future work is aimed at testing to prove accuracy of the approach, particularly in terms of equality with human cognition in certain domains.

7. RELATED WORK

The extraction of keywords challenges researches across many fields ranging from natural language processing to semantic technologies. There is no doubt that many applications will benefit from such algorithm. In the [12] a method for building ontology out of a domain-concerned vocabulary is proposed. The authors described how relevant terms are gathered by using documents judged representative of a given domain, by means of natural language processing. Once the necessary vocabulary is available, the connections among individual terms are being calculated, resulting in an ontology.

Similar approaches (e.g. [13]) were taken by several authors, and mostly rely on mining the knowledge from text resources by means of natural language processing.

The technique presented in this paper differs from existing techniques to some extent, since the Semantic Web methods are dealing with formally described data rather than ambiguous textual information. And our technique reflects this formal nature of data it processes.

8. CONCLUSIONS

In this paper we have presented the method (for vector description computation) that may be further exploited in vector driven evaluation of ontological models. As argued in [13], there is strong demand for an efficient evaluation technique and human-constructed *golden standard* is not good enough. The vector description driven evaluation may be employed in the very similar way the data driven evaluation is performed. The main contribution is in the ability of constructing a standard representation of a domain automatically, by using ontologies what makes this technique highly accurate in terms of ontological knowledge belonging to the domain.

The application, from our perspective, of such an approach is to create a reference entity of domains against which an ontology may be compared to establish a scope of a particular ontology. As mentioned, there are already few techniques for evaluation though these techniques struggle to find user base mainly because of drawbacks in the field of user interactions and usability (users comment on complexity whereas they expect something easy to work with and perform mostly simple tasks).

ACKNOWLEDGMENT

The work presented in this paper was supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within the 1/0042/10 project “Methods for identification, annotation, search, access and composition of services using semantic metadata in support of selected process types”.

REFERENCES

[1] FENSEL, D. – HENDLER, J. – LIEBERMAN, H. – WAHLSTER, W.: Spinning the Semantic Web, Cambridge, MA: MIT Press, 2006, pp. 3.

[2] d'AQUIN, M. – BALDASSARE, C. – GRIDINOC, L. – ANGELETOU, S. – SABOU, M. – MOTTA, E.: Watson: A Gateway for Next Generation Semantic Web Applications, Poster session of the International Semantic Web Conference, 2007, pp. 6–12.

[3] DING, L. – FININ, T. – JOSHI, A. – PAN, R. – COST, R. S. – YUN PENG, Y. – REDDIVARI, P. – DOSHI, V. – SACHS J.: Swoogle: A Search and Metadata Engine for the Semantic Web, Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management, 2004, pp. 1–8.

[4] DZBOR, M. – MOTTA, E. – BUIL ARANDA, C. – GOMEZ-PEREZ, J. M. – GOERLITZ, O. – LEWEN, H.: Developing ontologies in OWL: An observational study, Workshop on OWL: Experiences and Directions, 2006, pp. 2.

[5] DZBOR, M. – MOTTA, E. – BUIL ARANDA, C. – GOMEZ-PEREZ, J. M. – GOERLITZ, O. – LEWEN, H.: Analysis of user needs, behaviours & requirements wrt interfaces and navigation of ontologies, Deliverable report D4.1.1, NeOn Project, 2006, pp. 9–13.

[6] BRANK, J. – GROBELNIK, M. – MLADENIC, D.: A Survey of Ontology Evaluation Techniques, Department of Knowledge Technologies Jozef Stefan Institute, 2005, pp. 1–4.

[7] SALTON, G. – BUCKLEY, Ch.: Term-Weighting Approaches in Automatic Text Retrieval, 1988, pp. 2–3.

[8] FOX, C.: A stop list for general English, SIGIR Forum, 1989, pp. 19–35.

[9] ROCKAI, V.: Context for concepts, in 9th Scientific Conference of Young Researchers, 2009, pp. 1–2, ISBN 978-80-553-017B-5.

[10] MACHOVÁ, K. – BEDNÁR, P. – MACH, M.: Various approaches to web information processing, in Computing and Informatics, Vol. 26, No. 3, 2007, pp. 301–327, ISSN 1335-9150.

[11] LAU, K. – YIU-KAI, N.: A Client-Based Web-Cache Management System, Proceedings of the Third International Conference on Advances in Web-Age Information Management, 2002, pp. 3.

[12] NAVIGLI, R. – VELARDI, P. – CUCCHIARELLI, A. – NERI, F.: Automatic Ontology Learning: Supporting a Per-Concept Evaluation by Domanin Experts.Context for Concepts, In Proc. of Workshop on Ontology Learning and Population (OLP), in the 16th European Conference on Artificial Intelligence, 2004, pp. 2–4.

[13] BREWSTER, C. – ALANI, H. – DASMAHAPATRA, S. – WILKS, Y.: Data-driven ontology evaluation, in Proc. of the 4th International Conference on Language Resources and Evaluation, Lisbon, 2004, pp. 1–4.

Received February 21, 2011, accepted July 4, 2011

BIOGRAPHIES

Jozef Vrana was born on 27.06.1983. In 2001 he graduated (MSc) with distinction at the department of Cybernetics and Artificial Intelligence of the Faculty of Electrical Engineering and Informatics at Technical University in Košice. He defended his PhD thesis in 2010 in the field of the Semantic Web which remains his subject of interest.

Marián Mach graduated (MSc) in 1985 at the Department of Cybernetics and Artificial Intelligence at the Technical University in Košice. His PhD thesis on uncertainty processing in expert systems was defended in 1992. He is an associate professor at the Department of Cybernetics and Artificial Intelligence of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice. His scientific interests are knowledge management, data and web mining, classification of text documents, information retrieval, semantic technologies, and knowledge modelling.