Capturing Knowledge Representations Using Semantic Relationships

An Ontology-based approach

Ruben Costa, Paulo Figueiras, Luis Paiva, Ricardo Jardim-Gonçalves
Centre of Technology and Systems
UNINOVA
Quinta da Torre, Portugal
rddc@uninova.pt, paf@uninova.pt,
luismpaiva@mail.telepac.pt, rg@uninova.pt

Celson Lima
Federal University of Western Pará
PC / IEG / UFOPA
Santarém, Brasil
celsonlima@ufpa.br

***Abstract*— Knowledge representations in the scope of this work are a way to formalize the content of documents using dependent metadata i.e. words in document. One of the challenges relates to limited information that is presented in the document. While past research has made use of external dictionaries and topic hierarchies to augment the information, there is still considerable room for improvement. This work explores the use of complex relationships (otherwise known as Semantic Associations) available in ontologies with the addition of information presented in documents. In this paper we introduce a conceptual framework and its current implementation to support the representation of knowledge sources, where every knowledge source is represented through a vector (named Semantic Vector - SV). The novelty of this work addresses the enrichment of such knowledge representations, using the classical vector space model concept extended with ontological support, which means to use ontological concepts and their relations to enrich each SV. Our approach takes into account three different but complementary processes using the following inputs: (1) the statistical relevance of keywords, (2) the ontological concepts, and (3) the ontological relations.**

Keywords-Information Retrieval; Ontology Engineering; Knowledge Representation

I. INTRODUCTION

Knowledge and its respective representation has been part of human activity since immemorial times. Mankind created ways to tangibly represent sources of knowledge in order to preserve such knowledge and to guarantee that it would be transmitted to and reused by future generations. Classical examples are Egyptian papyrus and Sumerians clay tablets.

With the evolution of the World Wide Web towards the semantic web, knowledge sources (KS) and their representations have jumped on the main stage since they play a key role in this arena. Meaning of things and the ability to precisely understand them has been the holy grail of major efforts targeting the settlement (at least partial) of the tangible semantic web. Various sorts of concepts and tools have been developed and tested, the journey is very promising but there is a long way forward.

Controlled Vocabularies (CV) [1] have been considered good means to achieve this goal and, as such, a myriad of results & tools have been produced by researches around the world, based on the use of CVs. Among them, we are particularly interested in the use of ontological support to investigate the enrichment of knowledge representation of KS.

In this work, knowledge representation is expressed through the use of Semantic Vectors (SVs) based on the combination of the Vector Space Model (VSM) approach [2] and ontology-related features, namely ontological concepts and their semantic relations. Therefore, KS, in this work, are represented by SVs which contain concepts and their equivalent terms, weights (statistical, taxonomical, and ontological ones), relations and other elements used to semantically enrich each SV.

This paper is structured as follows. Section 2 defines the objectives and addresses the problem to be tackled. Section 3 presents the related work. Section 4 defines the process addressed by this work for knowledge representation. Section 5 illustrates the empirical evidences of the work addressed so far. Finally, section 6 concludes the paper and points out the future work to be carried out.

II. RELEVANCE OF THE PRESENTED WORK

This paper proposes the development of a framework to support the semantic representation of KS, which will be assessed in building and construction sector. Main features of this work include the analysis of the links among concepts, and the KS they are representing as well as the enhancement of such links with semantic relations among concepts.

In order to understand the importance of semantic relations within KS from the building and construction, one can think, for instance, on two expressions/terms (considered as ontological concepts, for the sake of clarity): “Design Phase” and “Architect”. These concepts are not father and son (hierarchically related), but they are inherently connected through a semantic relation described as “has Design Actor”, i.e., a project’s design phase may have many actors associated with it; one of them is the “Architect”. Such relation may also be associated to a given weight, i.e., how strong is the influence of the actor “Architect” within a project “Design Phase”.

Considering the example explained above, when a user is searching for information regarding a project design phase,

 ISBN: 978-1-61208-240-0

two different types of results may be expected by the end user, since "Design Phase" concept could be strongly related with the "Architect" concept.

The idea presented here is to enrich the representation of KS used/created within project teams on a collaborative engineering environment with information extracted from a domain ontology. A variety of semantic resources ranging from domain dictionaries to specialized taxonomies have been developed in the building and construction industry. Among them are BS6100 (Glossary of Building and Civil Engineering terms produced by the British Standards Institution); bcXML (an XML vocabulary developed by the eConstruct IST project for the construction industry); IFD (International Framework for Dictionaries developed by the International Alliance for Interoperability); OCCS (OmniClass Classification System for Construction Information) , BARBi (Norwegian Building and Construction Reference Data Library); and e-COGNOS (COnsistent knowledge management across projects and between enterprises in the construction domain). For the purpose of this work, a domain ontology was developed and validated in conjunction with the support of domain knowledge experts, and also adopting several concepts from the initiatives presented above. One of the reasons that lead a development of a new ontology, was due to the fact that at the time there was no support for OWL regarding such initiatives.

One of the novelties addressed by this work is the adoption of the Vector Space Model (VSM) approach combined with the ontological concepts and their semantic relations. The idea behind the VSM is to represent each document in a collection as a point in a space (a vector in a vector space). Points that are close together in this space are semantically similar and points that are far apart are semantically distant. The user's query is represented as a point in the same space as the documents (the query is a pseudo-document).

This approach uses an approximation to the VSM to achieve knowledge representations of documents and queries, and to define a relationship between these representations, allowing comparisons among them. The documents are sorted in order of increasing distance (decreasing semantic similarity) from the query and then presented to the user [3].

Knowledge representation of documents, using the VSM, often comes in the form of semantic vectors. Semantic vectors are usually called matrixes of frequencies, as they define the probabilistic frequency of the existence of a concept on a document and, hence, the relevance of that concept on the representation of the document.

III. RELATED WORK

In relation with the problem to be addressed by this work, Castells *et al*. [4] proposes an approach based on an ontology and supported by an adaptation of the Vector Space Model, just as in the presented work's case. It uses the TF-IDF (term frequency–inverse document frequency) algorithm [5], matches documents' keywords with ontology concepts, creates semantic vectors and uses the cosine similarity to compare created vectors. A key difference between this approach and the presented work is that Castells' work does not consider semantic relations or the hierarchical relations between concepts (taxonomic relations).

On the other hand, Nagarajan *et al.* [6] proposes a document indexation system based on the VSM and supported by Semantic Web technologies, just as in the presented work. They also propose a way of quantifying ontological relations between concepts, and represent that quantification in documents' semantic vectors. There are some differences between this work and the presented approach, which does not distinguish between taxonomic and ontological relations, as our approach does.

IV. THE PROCESS

The process being proposed by this work, is composed by several stages: the first stage (knowledge extraction) deals with the extraction of relevant words from KS, with the support of a text mining tool RapidMiner [7], and preforms a TF-IDF score for each relevant keyword within the corpus of KS that constitutes our knowledge base (knowledge sources repository); the second stage is the semantic vector creation, referred as Knowledge Source Indexation; and the third stage is document comparison and ranking processes, denominated Knowledge Source Comparison [8], as depicted in Figure 1.

The several stages that compose the process are illustrated with examples from a corpus with 70 KS related with the building and construction domain, where the creation of the sematic vectors example is described using an individual KS from the corpus.

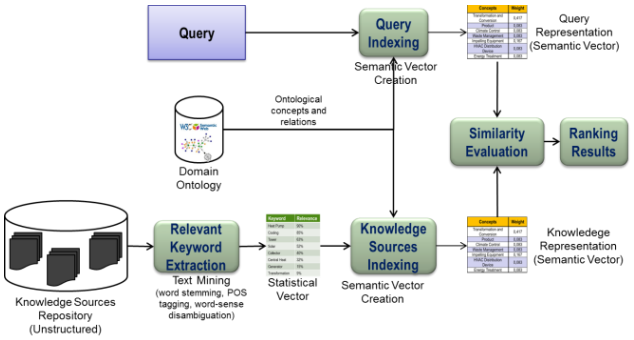

Figure 1. Document indexation and comparison

A. Knowledge Extraction

Although the use of text mining techniques is not the objective of this paper, it is worth to introduce some of text mining concepts, because the overall approach adopted here uses some of these concepts as an input to the knowledge representation mechanism.

Knowledge extraction is usually a process comprising three stages: word extraction, regular expressions filtering, and statistic vector creation.

Word extraction is the process in which words and expressions are extracted and divided through text-mining techniques. Regular expression filtering defines the process of removing expressions which have a great number of occurrences, but do not represent the knowledge within the

 ISBN: 978-1-61208-240-0

document (e.g. “and”, “the”, “when”). The last stage, statistic vector creation, is the process that builds the statistical representation of the documents in the form of a matrix composed by expressions, or keywords, and by the statistical weight of each keyword within the document, based on the TF-IDF score for each keyword within each KS.

Such structure is called statistical vector, and it is the main input for the presented work. Some frameworks and applications already treat knowledge extraction issues to the extent which our approach needs. Our approach uses RapidMiner to fulfil the needed knowledge extraction tasks and to create KS statistical vectors, which are then stored in a database.

It is important to mention that keywords presented in the statistical vector are composed by stemmed words (words that are considered a primitive form for a family of words, e.g. design: design, designer, designing, etc.). An example of such statistic vector for illustrative purposes is given in Table 1.

TABLE I. CONCEPTS AND WEIGHTS OF A DOCUMENT’S STATISTIC VECTOR

Keyword	Statistic weight (rounded values)
Agreement	0.550
Fund	0.376
Provis	0.317
Advanc	0.311
Record	0.250
Found	0.212
Feder	0.196
Local	0.166
Govern	0.153
Inspect	0.150
State	0.150
Ensur	0.144
Singl	0.116
modul	0.114
parti	0.114

B. Semantic Vector Creation

Semantic vector creation is the basis for the presented approach, it represents the extraction of knowledge and meaning from KS’s and the agglomeration of this information in a matrix form, better suited for mathematical applications than the raw text form of documents.

A semantic vector is represented by two columns: the first column contains the concepts that build up the knowledge representation of the KS, i.e. the most relevant concepts for contextualizing the information within the KS; the second column keeps the degree of relevance, or weight, that each term has on the knowledge description of the KS.

Our approach takes into account three different, but complementary procedures for building up the semantic vector, each of which is considered a more realistic iteration of the knowledge representation of a KS: Keyword-based, taxonomy-based and ontology-based semantic vectors.

Keyword-based semantic vectors are built upon the statistic representation of KSs in the form of expressions that occur in the document, according to their emphasis and frequency of occurrence both locally (in the KS itself) and globally (in the document corpus’ universe).

Table 2 depicts the weight of each ontology concept associated to each keyword within the statistic vector, where the first column corresponds to the ontology concepts that were matched to describe most relevant keywords extracted from the statistical vector, the second column indicates the most relevant keywords that were match to ontology equivalent terms, the third column corresponds the total ontology equivalent terms for each concept that was matched, and the fourth and last column, indicates the semantic weight for each ontology concept matched.

Taxonomy-based vectors push one notch further in the representation of KSs by adjusting the weights between expressions according to their taxonomic kin with each other, i.e., expressions that are related with each other with the “is a” type relation. If two or more concepts that are taxonomically related appear in a keyword-based vector, the existing relation can boost the relevance of the expressions within the KS representation.

Ontology-based vectors are the last iteration of the semantic vector creation process. The creation process for this type of vector uses the taxonomy-based vector as input to analyse the inherent ontological relation patterns between the input vector’s expressions. These ontological relations define semantic patterns between concepts which can be used to enhance the representation of the document. For instance, if a vector has two concepts that are related to each other by an ontological relation, and if this ontological relation occurs frequently across the document corpus’ universe, then the relevance of both concepts being together within the KS increases the weight of these concepts in the vector.

The major difference between taxonomy-based vectors construction and ontology-based vectors is that, taxonomy-based vectors take into account relations between concepts that are hierarchically related within the ontology tree (ex: father and son concepts). On the other hand, ontology-based vectors take into account relations between concepts that don’t need to be hierarchically related but are semantically connected. Examples of the two types of vectors are described in the following sub-sections.

TABLE II. CONCEPTS AND WEIGHTS OF A DOCUMENT’S STATISTIC VECTOR

Concept	Keyword	Ontology keywords	Sem. weight
Presence_Detection _And_Registration	record	recording	0.189
Foundation	found	foundation	0.134
Association	feder	federation	0.124
Inspector	inspect	inspector, inspection	0.114
Territory	state	state	0.095
Issue	compli	complicatio n	0.087
Trainer	manag	manager	0.028

 ISBN: 978-1-61208-240-0

Request	request	request	0.063
Consultant	author	authority	0.057
Management_Actor	manag	manager, management actor	0.028
Report	report	report	0.025

1) Keyword-based Semantic Vectors

The next step deals with matching the statistical vector's keywords with equivalent terms which are linked with the ontological concepts from the domain ontology. Equivalent terms for concept "Engineer" are shown in Figure 2. The matching process between equivalent terms presented on the domain ontology and the keywords within the statistical vector, is done by string matching. This approach may lead into some inconsistencies, since a keyword presented in the statistical vector may match two or more equivalent terms. This issue is being analysed and is considered to be part of future work.

It is worth also to mention that, the current process also addresses the introduction of new concepts and new semantic relations which are used to update the domain ontology. The process of updating the domain ontology is triggered every time new KS are introduced into the knowledge based. Algorithms for text processing (ex: association rules), are used to exploit new semantic relations between concepts or to update existing ones. This part of the process was intentionally not described here and is part of an on-going work.

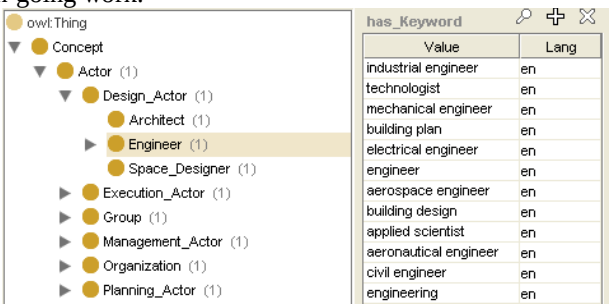

Figure 2. Ontological keywords and equivalent terms for concept "Engineer".

Each concept in the domain ontology has several keywords associated to it that present some semantic similarity or some meaning regarding that specific concept. Since keywords in the statistical vector comprise only stemmed words, several ontology-related keywords can be matched to one statistical vector's keyword. Although this fact may lead to some inconsistencies in terms of knowledge reliability, in this case, and because the presented work uses a very specific domain, these issues are decreased and are to be tackled in the future work section.

For each ontological concept that was extracted, the weights of all keywords matched with that concept are summed in order to get the total statistical weight for that ontological concept.

The next step to be performed, deals with the attribution of semantic weights to each of the concepts. The presented approach uses an approximation to the TF-IDF family of weighting functions [9], already used on other research works [4], to calculate the semantic weight for each concept resultant from the concept extraction process. The TF-IDF algorithm used is given by the expression:

$$w_x = \frac{w_{x,d}}{max_y w_{y,d}} \cdot \log \frac{\mathcal{D}}{n_x} \quad (1)$$

In Equation 1, $w_{x,d}$ is the statistical weight for concept x in KS d's statistical vector, $max_y w_{y,d}$ is the statistical weight of the most relevant concept, y, within the statistical vector of KS d, D is the total number of KSs present in the KSs search space, n_x is the number of KSs available in such space which have concept x in their semantic vectors, and w_x is the resultant semantic weight of concept x for document d.

Statistical normalisation is performed over the keyword-based semantic vector's weights, in order to obtain values between zero (0) and one (1).

This will be crucial for the upcoming vector comparison result ranking processes, because it will ease the computation processes needed and the attribution of relevance percentage to the results.

The keyword-based semantic vector is then stored in the database in the form $\left[\sum_{i=1}^{n} x_i \; ; \; \sum_{i=1}^{n} w_{x_i}\right]$, where n is the number of concepts in the vector, x_i is the syntactical representation of the concept and w_{x_i} is the semantic weight corresponding to concept.

2) Taxonomy-based Semantic Vectors

The taxonomy-based semantic vector creation process defines a semantic vector based on the relations of kin between concepts within the ontological tree. Specifically, the kin relations can be expressed through the following definitions [10]:

Definition 1: In the hierarchical tree structure of the ontology, concept A and concept B are homologous concepts if the node of concept A is an ancestor node of concept B. Hence, A is considered the nearest root concept of B, R(A,B). The taxonomical distance between A and B is given by:

$$d(A,B) = |depth(B) - depth(A)| = |depth(A) - depth(B)| \quad (2)$$

In Equation 2, depth(X) is the depth of node X in the hierarchical tree structure, with the ontological root concept's depth being zero (0).

Definition 2: In the hierarchical tree structure of the ontology, concept A and concept B are non-homologous concepts if concept A is neither the ancestor node nor the descendant node of concept B, even though both concepts are related by kin; If R is the nearest ancestor of both A and B, then R is considered the nearest ancestor concept for both A and B concepts, R(A,B); The taxonomical distance between A and B is expressed as:

$$d(A,B) = d(R,A) + d(R,B) \quad (3)$$

 ISBN: 978-1-61208-240-0

Figure 3 depicts the difference between homologous and non-homologous concepts.

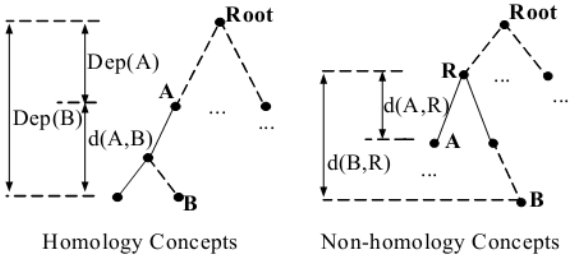

Figure 3. Homologous and non-homologous concepts (Li, 2009).

One of the major differences between our work and the work presented by Li [10], is that in our approach, the taxonomical weights between two concepts are not only related by their distance on the domain ontology, but also considering the relevance of the pair concepts A and B to each particular KS, i.e., if concepts A and B which are taxonomical related co-occur frequently, the taxonomical weight of such relation will be assigned a higher score.

3) Ontology-based Semantic Vectors

Other iteration of the semantic vector creation process is the definition of the semantic vector based on the ontological relations' which are defined in the domain ontology. Our system uses human input (knowledge experts in the building and construction domain) to establish final numerical scores on semantic relationships. The idea behind having a human intervene here is to let the importance of relationships reflect a proper knowledge representation requirement at hand. If the end-user is not interested in relationships between a project design phase and an architect actor, he should be able to rank those lower compared to other relationships. As an example, five ontological relations are shown in Table 3.

The first step is to analyse each ontological relation between concepts present on the input semantic vector. In this case, both keyword and taxonomy-based semantic vectors are used as inputs for this analysis. As in taxonomy-based semantic vector creation, there are two processes involved on the ontological relationship analysis: the first boosts weights belonging to concepts within the input semantic vector, depending on the ontology relations between them; the second adds concepts that are not present in the input vector, according to ontological relations they might have with concepts belonging to the vector [6].

In the first process (ontological relation between two concepts present in the input semantic vector), $\text{Co-Occurrence}_{C_xC_y}$ is computed with Equation 4, but this time it will be taken into account the frequency of occurrence of the ontologically related concepts throughout the document corpus.

$$\text{Co} - \text{Occurrence}_{C_xC_y} = idf(C_xC_y) = \log\frac{\mathcal{D}}{n_{C_xC_y}} \quad (4)$$

It is worth to notice, that an IDF calculus is performed but taking into account the ontological relation, i.e, the frequency of such relation is calculated within the all document corpus.

As in taxonomy-based semantic vector creation, the new concept is added to the semantic vector only if the ontological relation importance is greater than or equal to a pre-defined threshold, for the same constraint purposes. The ontological relation's importance, or relevance, is not automatically computed; rather, it is retrieved from an ontological relation vector which is composed by a pair of concepts and the weight associated to the pair relation.

In the case of the second process (ontological relation between one concept within the input semantic vector and another concept not comprised in that vector), and again as in the taxonomy-based semantic vector creation process, $C_x$is not modified and $C_y$is added to the semantic vector, and its weight is computed as in Equation 5.

$$tw_{C_y} = w_{C_y} + \sum(all\ related\ C_xs)\left[w_{C_x} * \left(TI_{C_xC_y}\right)\right] \quad (5)$$

TABLE III. EXAMPLES OF ONTOLOGICAL RELATIONS WITHIN ONTOLOGY.

Property	Subject	Object	Description
operates in	Actor	Project Phase	Actors operate in one or several particular project phases
is involved in	Actor	Project	Actors are involved in projects
has skills	Actor	Skill	Actors have some skills and expertise
has skill needs	Project	Skill	Projects need actors' skills and expertise
is decomposed in	Project	Task	Projects may be considered sets of tasks

V. ASSESSMENT

This section illustrates the assessment process of our approach. Firstly, the knowledge source indexation process will be assessed. Secondly, an example of a query and its results is exemplified.

A. Treating Queries

As mentioned earlier, queries are treated like pseudo-KSs, which means that all queries suffer an indexation process similar to the one applied to KSs. Initially, the query is divided into keywords and those keywords are then used to create a statistic vector for the query, equal to the statistic

term-frequency vector used for KS indexation. But, instead of passing the query through the knowledge extraction process the statistic vector is created by giving the same statistic weight to all keywords contained in the query. Such rule implies that the system assumes the same importance to all of the query's keywords.

For the purpose of this assessment, it was used a corpus of sixty five KS randomly selected, but all having a strong focus on the building and construction domain. Just as an example, a test query search for "door", "door frame", "fire surround", "fireproofing" and "heating", meaning that the user is looking for doors and respective components that are fireproof or that provide fire protection. In this case, keyword "door" is matched with concept "Door", "door frame" is matched with "Door Component", and so on, as shown in Table 5. Weights for matched ontological concepts are all equal to 0.2, because each concept only matches with one keyword. Hence, the semantic vector for this query will be the one of Table 4.

TABLE IV. EXAMPLE OF A QUERY'S SEMANTIC VECTOR.

#	Keyword	Ontology concept	Weight
1	Door	Door	0.2
2	door frame	Door Component	0.2
3	fire surround	Fireplace And Stove	0.2
4	Fireproofing	Fireproofing	0.2
5	Heating	Complete Heating System	0.2

B. Comparing and Ranking Documents

Our approach for vector similarity takes into account the cosine similarity [11] between two vectors, i.e. the cosine of two vectors is defined as the inner product of those vectors, after they have been normalized to unit length. Let d be the semantic vector representing a document and q the semantic vector representing a query. The cosine of the angle θ between d and q is given by:

$$\cos\theta = \frac{\mathrm{d}}{\|\mathrm{d}\|} \cdot \frac{\mathrm{q}}{\|\mathrm{q}\|} = \frac{\sum_{\mathrm{k}=1}^{\mathrm{m}} \mathrm{w_{dk}} \mathrm{w_{qk}}}{\sqrt{\left(\sum_{\mathrm{k}=1}^{\mathrm{m}} \mathrm{w_{dk}^2}\right)\left(\sum_{\mathrm{k}=1}^{\mathrm{m}} \mathrm{w_{qk}^2}\right)}} \quad (6)$$

where m is the size of the vectors, w_{dk} is the weight for each concept that represents d and w_{qk} is the weight for each concept present on the query vector q [4], [10].

A sparse-matrix multiplication approach is used because the most commonly used similarity measures for vectors d and q, such as cosine, can be decomposed into three values: one depending on the nonzero values of d, another depending on the nonzero values of q, and the third depending on the nonzero coordinates shared both by d and q.

In this case, calculating $f_1(d_k, q_k)$ is only required when both vectors have at least one shared nonzero coordinate. If the vectors do not possess any shared concept, i.e. a nonzero coordinate, the value for the function above is zero, and the vectors do not present any similarity. This also means that f2 and f3 do not need to be calculated, significantly reducing the computation needed [3].

On the other hand, even though the cosine method requires that both vectors have the same size, when using sparse-matrix multiplication the vectors' sizes do not necessarily have to coincide. If one vector is smaller than the other, then it means, in practice, that the smaller vector has zero values for all the concepts that are missing to reach the size of the bigger vector.

KS ranking is based on the similarity between KSs and the query. More specifically, and because the result of the cosine function is always 0 and 1, the system extrapolates the cosine function result as a percentage value.

The first result for the KSs tested is very satisfactory: the first search-resultant KS gives a relevance of 84% to the query, out of a total of sixty five KSs. The relevance of the KS corpus representation against the user query is presented in Table 5.

TABLE V. FIVE MOST RELEVANT RESULTS FOR THE USER QUERY.

Doc. Id	1	2	3	4	5	Query relevance%
190	0.093	0.093	0.077	0.077	0.0803	84
179	0.181	0.182	n.a.	n.a.	n.a.	57
201	0.121	0.122	0.013	0.013	n.a.	55
197	0.017	0.017	0.109	0.110	n.a.	52
172	0.045	0.045	0.035	0.037	0.012	48

It is easily comprehensible that, for the first result (doc. id 190), all concepts have higher semantic weight, with values near to 0.10 (or 10%). The second result presents high weights for the first two concepts, which means that it can have some relevance to the query, but its semantic vector does not contain the other three concepts of the query. This means that, although this KS has a good semantic reference to "Door" and "Door Component", it does not have knowledge about the other three concepts. The last result, with 48%, has weights for all concepts of the query but they are very low (4% maximum). This means that although the KS might have some relevance to the query, after a manual inspection over KSs tested, the results reflect knowledge contained within such documents.

The results are presented by showing only the relevance percentage for each KS the database identifier of the KS and the name and type of the KS file.

VI. CONCLUSIONS AND FUTURE WORK

Our contribution targets essentially the representation of KS which can be applied in various areas, such as semantic web, and information retrieval. Moreover, it can also support project teams working in collaborative environments, by helping them to choose relevant knowledge from a panoply of KS and, ultimately, ensuring that knowledge is properly used and created within organizations.

The results achieved so far and presented here do not reflect the final conclusion of the proposed approach and are part of an on-going work that will evolve and mature over

 ISBN: 978-1-61208-240-0

time, nevertheless preliminary results lead us to conclude that the inclusion of additional information available in domain ontologies in the process of representing knowledge sources, can augment such knowledge representations. Additional testing needed to be addressed, and other metrics for evaluating the performance of the proposed method (ex: precision and recall) needed to be implemented, in order to provide more concrete conclusions.

As future work, some improvements to the proposed approach within this work still needed to be carried out. As explained earlier, the corpus of KSs chosen to perform the assessment was adopting a randomly criteria. The fact that all documents are dealing with building and construction projects, make the scope very wide, which lead to a high level of noise introduced when creating statistical vectors adopting the TF-IDF approach. It is proposed as future work, to perform the creation of statistical vectors using a batch mode, where all KSs are previously grouped in clusters of domain area using clustering algorithms as the k-means algorithm. We believe that having documents previously grouped within clusters will reduce the level of noise introduced within the creation of statistical vectors.

Other operations for better enhance the semantic vectors can also be taken into account, for instance, union operations between taxonomical and semantic based vectors can also be seen as an approach for better represent KSs.

Additional work can also be driven on the building and construction domain ontology itself, which deals with the semantic features on knowledge representations. The domain ontology is seen as something that is static and doesn't evolve over time as organizational knowledge does. One possible approach to be adopted is to extract new knowledge coming from KSs (new concepts and new semantic relations) and reflect such new knowledge on domain ontology. The weights of such semantic relations should also be updated every time new KSs are introduced into the knowledge base. The idea is that, ontological concepts and relations should be inserted and managed dynamically, through a learning process, in order to make possible for the ontology to learn, capture mew concepts and relations from the KS corpus' universe and update relation importance between concepts, while new sources become available.

REFERENCES

[1] C. Lima, A. Zarli, and G. Storer, "Controlled Vocabularies in the European Construction Sector: Evolution, Current Developments, and Future Trends," Complex Systems Concurrent Engineering ed.London : Springer, pp. 565-574, 2007.

[2] G. Salton, A. Wong, and S. Yang, "A Vector Space Model for Automatic Indexing," Communications of the ACM, 18(11), pp. 613-620, 1975.

[3] P. Turney, and P. Pantel, "From Frequency to Meaning: Vector Space Models of Semantics," Journal of Artificial Intelligence, pp. 141-188, 2010.

[4] P. Castells, M. Fernández, and D. Vallet, "An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval" IEEE Transactions on Knowledge and Data Engineering, February, 19(2), pp. 261-272, 2007.

[5] T. Yu and G. Salton, G, "Precision weighting—An effective automatic indexing method," J. ACM 23, 1, 76–88, 1976.

[6] M. Nagarajan, A. Sheth, M. Aguilera, K. Keeton, A. Merchant and M. Uysal, "Altering Document Term Vectors for Classification - Ontologies as Expectations of Co-occurrence," ReCALL, p. 1225, 2007.

[7] Rapid-I GmBH, (2012, April). Retrieved from http://rapid-i.com/content/view/181/190/

[8] R. Costa, C, Lima, "An Approach for Indexation, Classification and Retrieval of Knowledge Sources in Collaborative Environments", Lisbon, 2011.

[9] S. Jones, "A Statistical Interpretation of Term Specificity and its Application in Retrieval," Journal of Documentation, 60(5), pp. 11-21, 2004

[10] S. Li, "A Semantic Vector Retrieval Model for Desktop Documents. Journal of Software Engineering & Applications," Issue 2, pp. 55-59, 2009.

[11] M. Deza and E. Deza, "Encyclopedia of Distances," Heidelberg: Springer-Verlag Berlin Heidelberg, 2009.

 ISBN: 978-1-61208-240-0