Information Retrieval in Collaborative Engineering Projects
A Vector Space Model Approach

Paulo Figueiras[1], Ruben Costa[1,2], Luis Paiva[2], Celson Lima[3] and Ricardo Jardim-Gonçalves[1,2]
[1]*UNINOVA, Centre of Technology and Systems, Campus da Caparica, Quinta da Torre, 2829-516 Monte Caparica, Portugal*
[2]*Faculade de Ciências e Tecnologia, Universidade Nova de Lisboa, Quinta da Torre, 2829-516 Monte Caparica, Portugal*
[3]*UFOPA / IEG / PSI, Federal University of Western Pará, Santarém, Brazil*

Keywords: Information Retrieval, Ontology Engineering, Knowledge Representation.

Abstract: This work introduces a conceptual framework and its current implementation to support the classification and discovery 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. SVs are compared against each other, in order to obtain their similarity index, and better support end users with a search/retrieval of knowledge sources capabilities. This paper presents the technical architecture (and respective implementation) supporting the conceptual framework, emphasizing the SV creation process. Moreover, it provides some examples detailing the indexation process of knowledge sources, results achieved so far and future goals pursued here are also presented.

1 INTRODUCTION

The World Wide Web has had a tremendous impact on society and business in just a few years by making information instantly and ubiquitously available. During this transition from physical to electronic means for information transport, the content and encoding of information has remained natural language. Today, this is perhaps the most significant obstacle to streamlining business processes via the web. In order that processes may execute without human intervention, documents must become more machine understandable.

The Semantic Web (Berners-Lee et al., 2001) is a vision of a future web of machine-understandable documents and data. On a machine understandable web, it will be possible for programs to easily determine what documents are about. For instance, people, places, events, and other entities that a document mentions will be canonically annotated within it.

This work addresses a knowledge representation approach that enables the user to express his information needs in terms of keywords, but at the same time uses the semantic information regarding the domain of the application to obtain results that are not possible in traditional searches. In traditional searches, a document is usually retrieved when at least one of the keywords in the query string occurs within it. The approach here is to obtain all concept instances that are related to a given word even if that word does not appear inside the concept.

One of the novelties of presented work is not only to analyse the relatedness between concepts, and ultimately, documents, but also to enhance such relatedness using semantic relations between concepts.

The idea presented here is to enrich the representation of knowledge sources, related with the building and construction sector, using a domain ontology with concepts and relations related with the construction sector. One of the novelties addressed by this work is the adoption of the Vector Space Model (VSM) (Salton et al., 1975) approach combined with the ontological concepts and their semantic relations represented by the domain ontology.

Figueiras P., Costa R., Paiva L., Lima C. and Jardim-Gonçalves R..
Information Retrieval in Collaborative Engineering Projects - A Vector Space Model Approach.
DOI: 10.5220/0004139302330238
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2012), pages 233-238
ISBN: 978-989-8565-30-3

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.

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

2 RELATED WORK

In relation with the problematic to be addressed by this work, (Castells et al., 2007) proposes an approach based on a ontology and supported by an adaptation of the VSM, just as in the presented work's case. It also uses the TF*IDF algorithm, matches documents' keywords with ontology concepts, creates semantic vectors and uses the cosine similarity to compare created vectors. A major difference between this approach and the presented work is that semantic relations are not considered, nor the hierarchical relations between concepts (taxonomic relations).

(Li, 2009) presents a way of mathematically quantifying such hierarchical or taxonomic relations between ontology concepts, based on relations' importance and on the co-occurrence of hierarchically related concepts, and reflect this quantification in documents' semantic vectors. This work's aim is to create an Information Retrieval (IR) model based on semantic vectors to apply over personal desktop documents, and has no relation to Web IR applications, as is the case of the presented work. Nevertheless, this work addresses some manual operations, where this work tries to automate.

(Nagarajan et al., 2007) propose 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 is a major difference between this work and the presented approach, though: (Nagarajan, et al., 2007) does not distinguish between taxonomic and ontological relations, as the presented approach does.

3 PROCESS

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

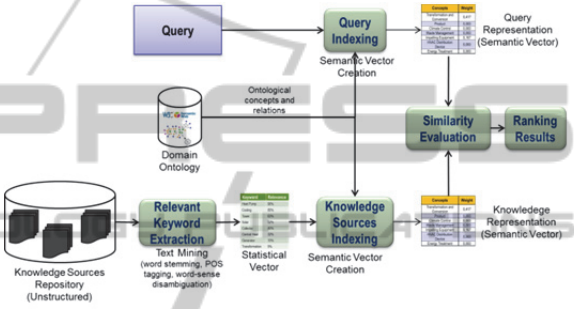

Figure 1: Document indexation and comparison process.

3.1 Knowledge Extraction

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

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 frequency and place of the keyword in it, as presented in Table 1.

Some frameworks and applications already treat knowledge extraction issues to the extent which our approach needs. This approach uses RapidMiner (Rapid-I GmBH, 2011) to fulfil the needed knowledge extraction tasks and to create documents' statistical vectors, which are then stored in a database.

An example of such statistic vector for a test document is given in Table 1. The presented values are low, due to the nature of knowledge sources: analysed knowledge sources reflect different topics under the Building and Construction domain.

Table 1: Concepts and weights of a document's statistic vector (incomplete).

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
…	…

3.2 Semantic Vector Creation

Semantic vector creation is the basis for the presented approach, it represents the extraction of knowledge and meaning from documents 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 as a matrix with two columns: The first column contains the concepts that build up the knowledge representation of the document, i.e. the most relevant concepts for contextualizing the information within the document; the second column keeps the degree of relevance, or weight, that each term has on the knowledge description of the document.

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

3.2.1 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 presented in the domain ontology. Equivalent terms for concept "Engineer" are shown in Figure 2.

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. This issue will be further analysed 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.

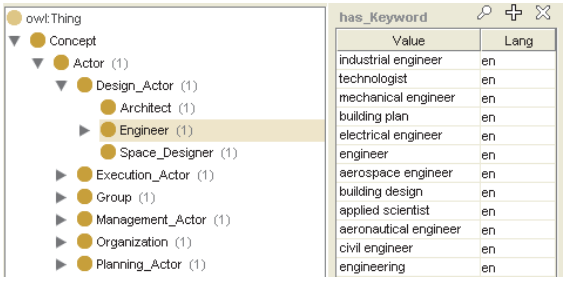

Figure 2: Ontological keywords and equivalent terms for concept "Engineer".

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 (Jones, 1972), already used on other research works (Castells et al., 2007), 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 document d's statistical vector, $max_y w_{y,d}$ is the statistical weight of the most relevant concept, y, within the statistical vector of document d, $\mathcal{D}$ is the total number of documents present in the documents' search space, n_x is the number of documents present in such search space which have concept x in their semantic vectors, and w_x is the resultant semantic weight of concept x for document d.

Statistical normalization is performed 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.

Table 2: Example of a keyword-based semantic vector (incomplete): Matched ontology concepts and keywords, and respective weights.

Concept	Keyword	Ontology keywords	Sem. weight
Presence_Detection_And_Registration	record	recording	0.189
Foundation	found	foundation	0.134
Association	feder	federation	0.124
Territory	state	state	0.095
Issue	compli	complicati on	0.087
Request	request	request	0.063
Consultant	author	authority	0.057
…	…	…	…

The keyword-based semantic vector is then stored in the database in the form $[\sum_{i=1}^{n} x_i \; ; \sum_{i=1}^{n} w_{x_i}]$, 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. The creation process for keyword-based semantic vectors is represented in Table 2 for the same test document used before.

3.2.2 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 (Li, 2009):

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.

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.

As referred before on this section, if two or more concepts are taxonomically related, this underlying relation may trigger two different processes (Nagarajan et al., 2007):

- *Process 1: When C_x (an ontology concept that belongs to the semantic vector) is taxonomically related to C_y (another ontology concept), and C_y is also present on the semantic vector.*

In this case, the weights corresponding to C_x and C_y are boosted within the semantic vector. Such weight boost is only performed if the taxonomic relation's importance is greater or equal than a certain threshold. This constraint only accepts the weight boost if the two related concepts are linked by a relation that is strong (i.e. both concepts are taxonomically near in the ontology tree).

Afterwards, the semantic vector's weights have to be normalized again, so that each weight represents a percentage of relevance on the knowledge representation of a document again.

- *Process2: When C_x (an ontology concept that belongs to the semantic vector) is taxonomically related to C_y (another ontology concept), and C_y is not present on the semantic vector.*

In this case, C_x is not modified and C_y is added to the semantic vector. The system has to calculate the TF*IDF weight for concept C_y, w_{C_y}, which brings a conceptual problem: C_y does not possess any statistic weight resulting from the Knowledge Extraction process. The chosen approach in this case is to apply only the IDF term of Equation 1. As in the previous process, the new concept is only added to the taxonomy-based semantic vector if the taxonomic relation's relevance is greater or equal than a threshold. For the example document, the concepts that were not present in the input semantic vector but had some taxonomical relation with concepts within such vector are presented in bold in Table 3, along with their respective weights (values are rounded).

Table 3: Old keyword-based weights and new taxonomy-based weights for the test document (incomplete).

Concept	Keyword weight	Taxonomy weight
Design_Actor	n.a.	**0.271**
Distributor	n.a.	**0.105**
Presence_Detection_And_Registration	0.189	0.095
Foundation	0.134	0.067
Contractor	n.a.	**0.063**
Association	0.124	0.062
Coordinator	n.a.	**0.062**
Inspector	0.114	0.057
Territory	0.095	0.048
…	…	…

It is important to notice that the threshold for considering both homologous and non-homologous relations was decreased, for the sake of clarity within this example, to visualize the concepts that were "boosted" and also new concepts that were included by the knowledge enrichment processes.

It is obvious that "*Design_Actor*" gained more relevance than all the other concepts, because the concept "*Design_Actor*" has a strong taxonomic relation with (i.e. is taxonomically near to) several concepts.

3.2.3 Ontology-based Semantic Vectors

Other iteration of the semantic vector creation process is the definition of the semantic vector based on the ontological relations' patterns present in the documents corpus.

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 (Nagarajan et al., 2007).

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.

4 ASSESSMENT

This chapter illustrates the assessment process of the proposed approach within this work. First, the knowledge source indexation process will be assessed. And finally, an example of a query and its results is exemplified.

4.1 Treating Queries

As mentioned before, queries are treated like pseudo-documents, which means that all queries suffer an indexation process similar to the one applied to documents.

For the purpose of this assessment, it was used a corpus of sixty five knowledge sources 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*" is inserted in the interface's keyword-based search field, 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 4. 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 4: 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

4.2 Comparing and Ranking Documents

Our approach for vector similarity takes into account the cosine similarity (Deza and Deza, 2009) between two vectors, i.e. its cosine, which is calculated by the Euclidian dot product between two vectors, and the sparse-matrix multiplication method, which is based on the observation that a scalar product of two vectors depends only on the coordinates for which both vectors have nonzero values.

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:

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

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 (Castells et al., 2007) (Li, 2009).

A sparse-matrix multiplication approach is adopted here, such as the cosine similarity, because the is one of the most commonly used similarity measures for vectors d and q and it 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.

Document ranking is based on the similarity between documents 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 results for the documents' test set is very satisfactory: The first search-resultant knowledge source presents a relevance of 84% to the

query, out of a total of sixty five documents. The relevance of the document corpus representation against the user query is presented in Table 5.

Table 5: 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%). Furthermore, all concepts of the query are contained in the first result's semantic vector.

5 CONCLUSIONS AND FUTURE WORK

Our contribution targets essentially the representation of knowledge sources 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 knowledge sources and, ultimately, ensuring that knowledge is properly used and created within organizations. Using semantic information from external ontologies is proven to enhance the representation of knowledge sources, but there is still some space for future improvements.

As future work, some improvements to the proposed approach within this work still needed to be carried out. It is proposed as future work, to perform the creation of statistical vectors using a batch mode, where all documents are previously grouped in clusters of domain area using clustering algorithms as the k-means algorithm.

Additional work can also be driven in order to apply learning mechanisms into the domain ontology. The domain ontology is seen as something that is static and doesn't evolve over time as organizational knowledge does. One possible approach is to extract new knowledge coming from knowledge sources (new concepts and new semantic relations) and reflect it on the domain ontology.

The idea of capturing user context (user profiling, type of platform being used by the end user, background information on past projects, etc.) in order to better enhance user experience and searching and ranking process of knowledge sources and must also be taken into account.

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.

REFERENCES

Berners-Lee, T., Hendler, J. & Lassila, O., 2001. The Semantic Web. *Scientific American,* pp. 34-43.

Castells, P., Fernández, M. & Vallet, D., 2007. 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.

Deza, M. M. & Deza, E., 2009. *Encyclopedia of Distances.* Heidelberg: Springer-Verlag Berlin Heidelberg.

Jones, K. S., 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation,* 28(1), pp. 11-21.

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

Nagarajan, M. et al., 2007. Altering Document Term Vectors for Classification - Ontologies as Expectations of Co-occurrence. *ReCALL,* p. 1225.

Rapid-I GmBH, 2011. *RapidMiner.* [Online] Available at: http://rapid-i.com/content/view/181/190/ [Acedido em 2011].

Salton, G., Wong, A. & Yang, C. S., 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM,* 18(11), pp. 613-620.