ONTOSPREAD: An API to Manage the Activation of Concepts in Graph-based Structures through the Spreading Activation technique

Jose María Álvarez[1], Diego Berrueta[2], Luis Polo[2], and José Emilio Labra[1]

[1] WESO RG, Universidad de Oviedo, Oviedo, Asturias, Spain,
{josem.alvarez,jelabra}@weso.es,
WWW home page: http://www.weso.es
[2] Fundación CTIC, Gijón, Asturias, Spain,
{diego.berrueta,luis.polo}@fundacionctic.org,
WWW home page: http://www.fundacionctic.org

Abstract. The present article introduces the ONTOSPREAD API for the development, configuration, customization and execution of the Spreading Activation technique over graph-based structures more specifically over semantic networks, RDF graphs and ontologies arising from the Semantic Web area. This technique has been used to the efficient exploration and querying of large and heterogeneous knowledge bases based on semantic networks in the Information or Document Retrieval domains. ONTOSPREAD implements the double process of activation and spread of concepts in ontologies, implicit graph structures, applying different restrictions of the original model like weight degradation according to the distance or others coming from the extension of this technique like the converging paths reward. The main application of Spreading Activation lies in two different areas of interest to digital libraries: 1) construction of hybrid semantic search engines 2) ranking of information resources according to an input set of weighted resources. These techniques provide a whole framework to ease the information access, a common required features in the exploitation of new and existing digital libraries. Finally, in this work an evaluation methodology and an example using the Galen ontology are provided to validate the goodness, the improvement and the capabilities of this framework applied to digital libraries.

1 Introduction

The improvements in digitization lead us to a new environment in which digital libraries and archives are designed and used in a new way. This situation implies new challenges in the digital formats, storage (information is continuously growing) and information retrieval models. Following the recommendations of the European Commision [10] the digital libraries are a key factor to bring out the full economic and cultural potential of Europe's cultural and scientific heritage through the Internet. The online presence of material from different cultures and in different languages will make it easier for citizens to appreciate their own cultural heritage as well as the heritage of other European countries. Besides its fundamental cultural value, cultural material is an important resource for new added value services. That is why more sophisticated software tools and methods are needed to meet the expectations of users easing the information retrieval of these large datasets and overcoming the classical problems of information overloading.

Initiatives like Semantic Web and Linked Data [22] tries to define vocabularies and ontologies enabling the data interoperability and sharing that enable by means of the Web infrastructure the access to the contents across different platforms and applications. The development of tools using these common data formats and models is largely implemented and representative to digital libraries but some algorithms and methods are not yet promoted to work with them in a standard way preventing the improvement and effectiveness of information access.

In this sense Spreading Activation technique (hereafter SA) introduced by [9], in the field of psycho linguistics and semantic priming, purpose a model in which all relevant information is mapped on a graph as nodes with a certain "activation value". Relations between two concepts are represented by a weighted edge. If a node is activated their activation value is spread to their neighbor nodes. This technique was adopted by the computer science community and applied to the resolution of different problems, see Sect. 3. In the field of digital libraries this technique can ease the information access providing a connectionist method to retrieve data like brain can do. Although SA is widely used, more specifically in recent years have been successfully applied to ontologies, a common and standard API is missing and each third party interested in their application must to implement its own version [32] of SA.

Taking into account this new information realm and the leading features of putting together the SA and the Semantic Web technologies, new enriched services of searching, matchmaking, recommendation or contextualization can be implemented to fulfill the requirements of access information in digital libraries of different trending scopes like e-procurement, legal document databases [5], e-tourism or e-health.

1.1 Main Contributions

The proposed work aims to afford a framework for SA to ease the configuration, customization and execution of them over semantic networks and more specifically over RDF graphs and ontologies. It is relevant to digital libraries access and interoperability due to the fact that this technique provides a set of proven algorithms for retrieving and recommending information resources in large knowledge bases. Following the specific contributions of this work are listed:

- Study and revision of the classical constrained SA.
- Study and definition of new restrictions for SA applied to ontologies and RDF graphs.

- Implementation of a whole and extensible framework (called ONTOSPREAD API). to customize and perform the SA based on good practices in software programming.
- Outlining of a methodology to configure and refine the execution of SA.
- An example of configuration and refinement applying SA over a well-known ontology, the Galen ontology.

2 Background

In this section, the theoretical model of *SA* [9,3] is reviewed to illustrate the basic components and the operations performed by SA during their execution, specially the spread of the activation from a node to their adjacent, see Fig. 1. This model is made up of a conceptual network of nodes connected through relations (conceptual graph). Taking into account that nodes represent domain objects or classes and edges relations among them, it is possible to establish a semantic network in which SA can be applied. The process performed by the algorithm is based on a thorough method to go down the graph using an iterative model. Each iteration is comprised of a set of beats, a stepwise method, and the checking of a stop condition. Following the different stages of SA are presented and defined:

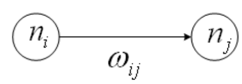

Fig. 1. Graphical model of *Spreading Activation*

***Preadjustement*:** This is the initial and optional stage. It is usually in charge of performing some control strategy over the target semantic network.

***Spreading*:** This is the spread stage of the algorithm. Concepts are activated in activation waves. The spreading node activates its neighbor nodes, see Fig. 2.

The calculation of the activation rank I_i of a node n_i is defined as follows:

$$I_i = \sum_j O_j \omega_{ji} \tag{1}$$

I_i is the total inputs of the node n_i, O_j is the output of the node n_j connected to n_i and ω_{ji} is the weight of the relation between n_j and n_i. If there is not relation between n_j and n_i then $\omega_{ji} = 0$.
The activation function f is used to evaluate the "weight" of a node and decide if the concept is active.

$$N_i = f(I_i) = \begin{cases} 0 & \text{if } I_i < \jmath_i \\ 1 & \text{if } I_i > \jmath_i \end{cases} \tag{2}$$

N_i is 1 if the node has been activated or 0 otherwise. $\jmath_i$, the threshold activation value for node i, depends on the application and it can change from a node to others. The activation rank I_i of a node n_i will change while algorithm iterates.

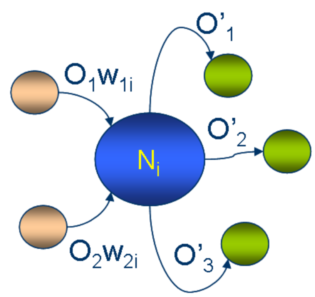

Fig. 2. Activation of concepts in *Spreading Activation*

***Postadjustment*:** This is the final and optional stage. As well as *Preadjustment* stage, it is used to perform some control strategy in the set of activated concepts.

3 Related Work

Since SA was introduced by [9] in the field of psycho linguistics and semantic priming it has been applied to the resolution of problems trying to simulate the behavior of the brain using a connectionist method to provide an "intelligent" way to retrieve information and data.

The use of SA was motivated due to the research on graph exploration [27,3,8]. Nevertheless the success of this technique is specially relevant to the fields of Document [23] and Information Retrieval [11,4,1,21]. It has been also demonstrated its application to extract correlations between query terms and documents analyzing user logs [12] and to retrieve resources amongst multiple systems [30] in which ontologies are used to link and annotate resources.

In recent years and regarding the emerging use of ontologies in the Semantic Web area new applications of SA have appeared to explore concepts [28,7] addressing the two important issues: 1) the selection and 2) the weighting of additional search terms and to measure conceptual similarity [20]. On the other hand, there are works [24,13] exploring the application of the SA on ontologies in order to create context inference models.The semi-automatically extension and refinement of ontologies [25] is other trending topic to apply SA in combination with other techniques based on natural language processing. Data mining, more specifically mining socio-semantic networks[33], and applications to collaborative filtering (community detection based on tag recommendations, expertise location, etc.) are other potential scenarios to apply the SA theory due to the high performance and high scalability of the technique.

In particular, annotation and tagging [17,18] services to gather meta-data [19] from the Web or to predict social annotation [6] and recommending systems based on the combination of ontologies and SA [16] are taken advantage of using SA technique.

Finally the semantic search [31,34] is a highlight area to apply SA following hybrid approaches [5,29] or user query expansion [26] combining metadata and user information.

Although this technique is widely accepted and applied to different fields open implementations, Texai[3] company offers a proprietary implementation of SA, are missing. Moreover the Apache Mahout [4] project, a recent scalable machine learning library that supports large data sets, does not include an implementation of SA instead of providing algorithms for the classification, clustering, pattern mining, recommendation and collaborative filtering of resources in which SA should be representative.

4 ONTOSPREAD API

In this section, the definitions of classical restrictions of SA and new extensions of the algorithm are provided. Afterwards the outstanding parts of the API are presented putting up a generic and extensible framework to the SA technique.

4.1 Constrained Spreading Activation

One of the leading features of SA technique is its flexibility to fit to the resolution of different kind of problems. From the configuration point of view some constraints presented in [8] have been customized to improve the expected outcomes of the execution according to the domain problem.

Distance: nodes far from an activated node should be penalized due to the number of needed steps to reach and activate them.
Path: the activation path is built by the activation process from a node to other and this process can be guided according to the weights of relations (edges).
Multiple outputs (Fan-Out): nodes "highly connected" can guide to a misleading situation in which activated and spread nodes are not representative, these nodes should be skipped or penalized by the algorithm.
Threshold activation: a node n_i will be spread iif its activation value, I_i, is greater than a threshold activation constant $\jmath$.

[3] http://texai.org/
[4] http://mahout.apache.org/

The aforementioned theoretical model is an excellent start point to design an API for *SA* but from the domain expert point of view some configuration requirements to apply this technique to ontologies are missing. That is why a set of extensions are proposed to deal with the specific features of ontologies and RDF graphs.

Context of activation $\mathbb{D}_{com}$: concepts can be defined in different domains or schemes. The double process of activation and spreading will only be performed in the context $\mathbb{D}_{com}$.

Definition 1. *Let $\mathbb{D}_{com}$ an active domain, if a concept c_i is activated o spread then $c_i \in \mathbb{D}_{com}$.*

Minimum activation value $N_{\min}$: only concepts with an activation value N_k greater than $N_{\min}$ will be spread. This constraint comes from the theoretical model of *SA*.

Maximum number of spread concepts $\mathbb{M}$: the process of activation and spreading will be performed, at the most, until $\mathbb{M}$ concepts had been spread.

Minimum number of spread concepts $\mathbb{M}_{\min}$: the process of activation and spreading will be performed, at least, $\mathbb{M}_{\min}$ concepts had been spread.

Time of activation t: the process of activation and spreading will be performed, at the most, during t units of time.

Output Degradation O_j: one of the keypoints to improve and customize the algorithm is to define a function h that penalizes the output value O_j of a concept c_j.

1. Generic customization: h calculates the output of a concept c_j according to its degradation level.

$$O_j = h(I_j) \tag{3}$$

   Basic case: if $h_0 = id$, the output value O_j takes the level of the activated concept c_j as its value.

$$O_j = h_0(I_j) = I_j \tag{4}$$

2. Customization using **distance**: h_1 calculates the level activation of the concept c_j according to the distance from the initial concept $c_l \in \Phi$[5] to the node that has activated it. The activation value should decrease if the distance from Φ grows thus the algorithm follows a path from c_l to c_j: $I_l > I_j$.

   The function h_1 penalizes the output of concepts (decreasing their rank) far from the "activation core" and rewards closed concepts. Thus, let d_j, where $d_j = min\{d_{lj} : \forall n_l \in \Phi\}$:

$$O_j = h_1(I_j, d_j) = \frac{I_j}{d_j} \tag{5}$$

[5] Set of initial concepts.

3. Customization using **beats**: the function h_2 calculates the degradation of the concept using the number of iterations k:

$$O_j = h_2(I_j, k) = (1 + \frac{I_j}{k}) \exp(-\frac{I_j}{k}). \tag{6}$$

4.2 Specification of the algorithm

The entry point to SA technique is the set of initial concepts (Q_{sem}) that will generate a new set of the most relevant concepts (Q'_{sem}). Ontologies based on the RDF graph model are a graph where each node n_i represents a concept c_i and the edge ω_{ji} is the semantic relation between c_j y c_i. The final result of the algorithm is a set of sorted pairs (n_i, I_i) that builds Q'_{sem}, where $n_i \approx c_i$ and $I_i \approx w_i$ (the relevance of the concept).

The implementation of SA, see Algorithm.1, comprises of two sets of concepts that store information about the state of the algorithm: 1) $\mathbb{D}_{com}$ are all the concepts in the semantic network and 2) Φ[6] is the set of initial activated concepts, c_j^k is the spreading concept at the k-esima iteration (from which other concepts are activated).

Set $\mathcal{A}$: queue of **activated** concepts (candidates to be spread).

$$\mathcal{A}^0 = \Phi \tag{7}$$

$$\mathcal{A}^k = (\mathcal{A}^{k-1} \cup \{c_i : \forall c_i / \omega_{ji}^k > 0\}) - \{\mathcal{G}^k\} \tag{8}$$

Set $\mathcal{G}$: set of spread concepts:

$$\mathcal{G}^0 = \emptyset \tag{9}$$

$$\mathcal{G}^k = \mathcal{G}^{k-1} \cup \{c_j^k\} \tag{10}$$

The output of the algorithm is the new enriched query $\mathcal{G}^k = Q'_{sem}$ made up of the set of weighted concepts.

Finally, the calculus of the activation value of a concept c_i at iteration k, indicated by I_i^k, is defined. At 0 iteration the activation value c_i is calculated as follows:

$$I_i^0 = \begin{cases} 1 & \text{si } c_i \in \Phi \\ 0 & \text{si } c_i \notin \Phi \end{cases} \tag{11}$$

at k iteration, the activation value of c_i from element c_j^k to c_i is calculated as follows:

$$I_i^k = \begin{cases} I_i^{k-1} & \text{si } \omega_{ji}^k = 0 \\ I_i^{k-1} + \omega_{ji}^k I_j^{k-1} & \text{si } \omega_{ji}^k > 0 \end{cases} \tag{12}$$

[6] $\Phi \equiv Q_{sem}$.

Algorithm 1 *Pseudocode of Spreading Activation*

Require: $\Phi \neq \emptyset$
Ensure: $\mathcal{G} \neq \emptyset$

```
  𝒜 ← Φ
  𝒢 ← ∅
  while 𝒜 ≠ ∅ AND card(𝒢) < 𝒢_min AND N_k ≥ N_min do
     n_k ← extract(𝒜)
     𝒢 ← {n_k} ∪ 𝒢
     for all n_i/w_ki > 0 do
        N_i ← N_i + w_ki N_k
        𝒜 ← ({n_i} ∪ 𝒜) − 𝒢
     end for
  end while
  return 𝒢
```

4.3 Improving Spreading Activation

Some improvements in the calculus of the activation value of a concept have been introduced in order to get a more complete and accurate technique. If some paths of activation converge to the same node, see Fig. 3 and the source nodes are different then this node should be relevant and a reward is applied to the nodes presented in these paths.

Definition 2. *Let p_i the number of paths that start and finish in different nodes of Φ[7] and they go through the node c_i and they only contain nodes belonging to $\mathcal{G}$. This improvement assigns a new value to the activation value of each node c_i indicated by I_i^* and it is calculated by means of the function g:*

$$I_i^* = g(I_i, p_i) \tag{13}$$

In this case, a relaxed reward function has been chosen, Eq. 14, and, it is applied in the *Postadjustment* stage, thus the original semantics and behavior of *SA* algorithm remains.

$$g(x, y) = x(\log(y + 1) + 1) \tag{14}$$

x is the reward constant, it can be defined according to the context and y is the number of times that a concept c_i must be rewarded.

4.4 Refining Spreading Activation

The whole configuration of the algorithm can be made by default but a customization to a particular domain should be carried out by a domain expert taking into account the specific issues of this domain and considering it as a new stage of the ontology o graph modeling process:

[7] The reward is not applied to nodes in Φ.

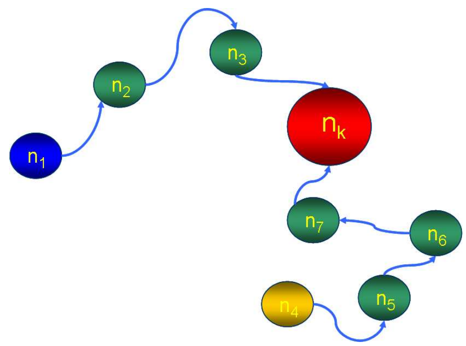

Fig. 3. Paths and rewards in *Spreading Activation*

1. The algorithm is highly coupled to the target ontology and domain. Thus the adjustment and customization of the algorithm should be created or supervised by experts with domain knowledge.
2. The establishment of relation weights among concepts is the key point to customize *SA* techniques.

Since *SA* uses weights in relations to calculate the activation value of the concepts, different "patterns", see Tab. 1, have been identified to manage the direction of the spreading process.

Spread Direction	**Definition**	**Key Relation**
Ascending	It seeks for the activation of concepts more generic than the current.	"superclass"
Descending	It seeks for the activation of concepts more specific than the current.	"superclass"
Nominal	It seeks for the activation of instances instead of concepts.	"instance of"
Crossing	It seeks for the activation of concepts and instances connected through a certain relation $\mathcal{R}$.	$\mathcal{R}$

Table 1. Patterns to manage the direction of SA.

These control patterns can be put together in order to fit as much as possible the focus and direction of the double process of activation and spreading.

4.5 Design and implementation of ONTOSPREAD API

ONTOSPREAD API is addressed by an open and extensible design applying best practices on software design [15,2] and development [**?**,14]. The next basic objects to implement *SA* technique have been identified.

– The *Player* class handles the execution of the algorithm in a stepwise way. This is an application of the *Iterator* design pattern to perform the activation and spreading processes. The state of the algorithm is captured in a separate class, *OntoSpreadState* thus it is possible to serialize the state and back to a previous one.
– The *SA* process comprises of three sub-processes:
  1. *Preadjustement*: *OntoSpreadPreAdjustment*; 2. *Spreading* (activation and spreading) with constraints: *OntoSpreadRun*; and 3. *Postadjustment*: *OntoSpreadPostAdjustment*.
  Moreover, the process carries on the information about the knowledge base using the DAO pattern thus the API is independent from the modeling language of the semantic network. Currently, OWL and RDF are supported.

Fig. 4. ONTOSPREAD Overview Diagram

4.6 Designing the state of *SA*

The keypoint to design the algorithm lies in how and where the information will be available at different iterations. Secondly, an unique entry point to the state of the algorithm should be available trying to avoid illegal accesses. This object stores the next information: 1. Spread concepts. 2. Active concepts. 3. Paths of activation. 4. Concept to be spread. 5. Generic swap area (to share information among iterations).

4.7 Designing the restrictions of *SA*

The extensibility and flexibility of the algorithm is subjected to a good design of the restrictions and their evaluation process. The next features and design patterns are used to design and implement the model of restrictions of SA:

– Any restriction can be considered as a simple restriction and can be evaluated to a boolean value.
– Conditions or actions in the algorithm can be comprised of several restrictions.
– The extension points of the algorithm, included through a *Template Method* design pattern, are strategies to carry out an specific action. Each strategy can be subjected to one or several restrictions.
– Each restriction can be simple or comprised of others. *Composite* pattern.
– Each action is an strategy. *Strategy* pattern.
– A strategy implies one restriction (or a set of them) thus the strategy is a client of the *Composite* of restrictions.

- The evaluation of the restrictions to get their value (boolean) is carried out through a *Visitor* pattern that fits perfectly to evaluate and walk in composite objects.
- The evaluation process consists on: apply the strategy that modifies the state of the algorithm and assert this change of state by means of the restrictions applied to this strategy.

4.8 Development of ONTOSPREAD API and Supporting Tools

The development of the API has been performed using Semantic Web and Java technologies like: Jena[8] API, JAXB[9], Maven[10] o Spring[11]. Besides two tools are provided to test and debug different configurations of the algorithm:

ONTOSPREAD-TEST It is a tool for the automatic execution and reporting of batch tests. It provides an user-oriented framework to configure, combine and load several configurations (e.g. restrictions, weights, initial concepts, etc.) for *SA* and get results. A XML vocabulary using XML-Schema and the *Extensible Content Model* XML design pattern has been defined to build the configuration of the *SA* process. The designing of this vocabulary is oriented to be used with JAXB, this technology enables us automatically the processes of marshalling and unmarshalling Java classes providing an easy way to configure, load and serialize the different configurations and results.

ONTOSPREAD Inspector. It is a graphical debugger of *SA* algorithm using the graph library JpowerGraph[12] and the SWT[13] toolkit. It enables to configure, load, run (one step or stepwise) and view the evolution of the semantic network with the concepts (activated, spread, weights, relations, etc.).

Finally, a set of source code metrics using Eclipse[14] and the metrics plugin[15] have been extracted to demonstrate and measure its quality. Table2 presents these metrics that measures cohesion and coupling of the software.

[8] http://jena.sf.net
[9] http://java.sun.com/developer/technicalArticles/WebServices/jaxb/
[10] http://maven.apache.org
[11] http://www.springframework.org/
[12] http://jpowergraph.sf.net
[13] http://www.eclipse.org/swt/
[14] http://www.eclipse.org
[15] http://metrics.sf.net/

Table 2: Source Code Metrics

Source Code Metrics						
ID	*Def.*	*Total*	*Avg.*	*Std. Dev.*	*Max*	*Scope*
TLOC	Total Lines of Code.	5272				
CA	Afferent Coupling.		6.524	10.545	48	Package.
RMD	Normalized Distance, $\|RMA + RMI - 1\|$.		0.32	0.347	1	Package.
NOM		0.065	0.296	3	Type.	
RMI	Instability: $CE/(CA + CE)$		0.567	0.387	1	Package.
NBD	Nested Block Depth.		1.204	0.516	4	Method.
LCOM	Lack of Cohesion of Methods (*Henderson-Sellers*).		0.18	0.289	0.957	Type.
VG	McCabe Cyclomatic Complexity.		1.297	0.735	6	Method.
RMA	Abstractness.		0.113	0.186	0.667	Package.
CE	Efferent Coupling.		1.976	1.282	5	Package.
DIT	Depth of Inheritance Tree.		1.607	0.915	4	Type.

All of these values are in the default range defined in the plugin thus the quality of the source code with the desired feautures of *high cohesion* and *low coupling* are assured.

5 Evaluation of ONTOSPREAD API

The validation of the algorithm depends on the configuration of the activation and spreading processes to fit it to the different domain issues. SA is determined by the target semantic network and therefore the domain knowledge (concepts and relations) defined is the key part to adjust its behavior. On the other hand, taking into account that the activation and spreading is guided by the weights of relations the specification of them is fundamental to get the desired outputs. The methodology to test the implementation of the algorithm is subjected to these conditions but a step-wise refinement method can be outlined:

1. Use a semantic network (ontology, etc.) well-known: concepts and relations.
2. Define potential set of initial concepts (Q_{sem}) and their initial activation value (commonly 1.0).
3. Specify the weights of the relations to that domain knowledge.
4. Combine the different restrictions provided by the API.
5. Select the degradation function.
6. Add the reward techniques to increase the activation value of certain nodes.
7. Try to evaluate new activation functions for their further implementation.

8. Repeat these steps until getting the most appropriated set of output concepts (Q'_{sem}) to that domain knowledge

To apply this methodology, the Galen Ontology [16] has been selected. It is the reference ontology in the biomedicine domain and it is widely used in reasoning and decision support processes. The design of the experiment depends on: the ontology (the API is not restricted to work with only one ontology), the weights of relations, the set of initial concepts, the set of restrictions, the degradation function and the extensions to reward nodes. Following these variables are specified to run the algorithm and get a good refinement of SA in the selected domain. The structure of the Galen ontology is summarized in Tab 3:

Table 3: Galen Ontology.

Stats of Galen Ontology	
Type	*Number*
Object Properties	0
Clases	0
Enumerated Classes	0
Union Classes	0
ComplementOf Classes	0
Intersection Classes	0
Restrictions	0
Instances	0

The set of initial concepts (Q_{sem}) with an initial value 1.0 is: "#Foo" and "#Bar". The weights of the relations are fixed to a default value of 1.0. The refinement of the algorithm will enable us to get a set of output concepts (Q'_{sem}) similar to the process that a brain will do. The degradation functions and the reward technique will be alternatively combined checking the output of the algorithm. Finally, the restrictions and their configuration values are listed in Tab. 4.

The execution of the first test get the next results, see Tab. 5:

Table 5: Results of the first test.

Output Metrics	
Type	*Value*
Spread Nodes	0
Activated Nodes	0
Highest activation value	0
Deepest spread path	∞
Time of activation t	0
Degradation function	h_1
...	

[16] http://www.opengalen.org/

Table 5: Results of the first test (continues).

Reward	No
Q'_{sem}	
URI	*Value*
#Foo	0
...	

After that the configuration is changed to use the reward of converging paths and adjusting the restrictions with the next values: FIXME: values. The results of this execution are demonstrated in Tab. 6

Table 6: Results of the second test.

Output Metrics	
Type	*Value*
Spread Nodes	0
Activated Nodes	0
Highest activation value	0
Deepest spread path	∞
Time of activation t	0
Degradation function	h_1
Reward	No
Q'_{sem}	
URI	*Value*
#Foo	0
...	

FIXME: Concluir esultados

6 Conclussions and Future Work

FIXME: Revisar

This work provides a configurable and extensible API to support the SA technique. It allows the configuration of restrictions and their combination to get the most accurate set of output concepts. One of the features that turns SA to a widely accepted algorithm lies in its flexibility but some disadvantages are also presented: the adjusting and refinement of the restrictions and the weights of the relations, the selection of the degradation function or the establishment of reward functions. This API minimizes these advantages with an extensible framework that can be applied to different scenarios like digital libraries in particular biomedicine, e-procurement, e-health, etc. providing enriched services of annotation, searching or recommendation.

The main improvement in the algorithm consists on the flexibilization of the refinement methodology. An automatic learning algorithm to create SA configu-

Restriction Values	
Type	*Value*
Minimum activation value $N_{\min}$	0
Minimum number of spread concepts $\mathbb{M}_{\min}$	0
Maximum number of spread concepts $\mathbb{M}$	0
Time of activation t	0
Context of activation $\mathbb{D}_{com}$	DEFAULT ($retries$=0)

Table 4. Restrictions to test SA.

rations according to ontologies should be developed. Thus, the training stage of SA could generate the best configuration for a specific domain. The algorithm could optimize the selection of input parameters like the weights of the relations, the degradation functions or the combination of restrictions. Beside new mesaures related to instances such as 'Cluster Measure", "Specifity Measure" or both could be used in the process of activation/spreading. Also the selection of the next node to spread is based on a "first better" strategy (if two nodes have the same activation value) because of this fact other selection strategies should be implemented.

Acknowledgements

ONTOSPREAD API has been initially developed in BOPA [5] project that is one of *Semantic Web Use Cases and Case Studies*[17] collected by W3C. Currently it is being applied to the process of searching public procurement notices in the "10ders Information Services"[18] project, partially funded by the Spanish Ministry of Industry, Commerce and Tourism (TSI-020100-2010-919) and the European Regional Development Fund (EFDR), leaded by "Gateway Strategic Consultancy Services"[19] and developed in cooperation with "Exis TI"[20] and WESO Research Group.

References

1. Maristella Agosti and Fabio Crestani. A Methodology for the Automatic Construction of a Hypertext for Information Retrieval. In *SAC '93: Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing*, pages 745–753, New York, NY, USA, 1993. ACM Press.

[17] http://www.w3.org/2001/sw/sweo/public/UseCases/CTIC/
[18] http://rd.10ders.net
[19] http://gateway-scs.es/
[20] http://www.exis-ti.com/

2. Deepak Alur, John Crupi, and Dan Malks. *Core J2EE Patterns: Best Practices and Design Strategies.* Sun Microsystems, 2003.
3. John R. Anderson. A Spreading Activation Theory of Memory. *Verbal Learning and Verbal Behavior*, (1):261–295, 1983.
4. Helmut Berger, Michael Dittenbach, and Dieter Merkl. An adaptive information retrieval system based on associative networks. In *CRPIT '04: Proceedings of the first Asian-Pacific conference on Conceptual modelling*, pages 27–36, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
5. Diego Berrueta, Jose Emilio Labra, and Luis Polo. Searching over Public Administration Legal Documents Using Ontologies. In *Proceedings of Joint Conferente On Knowledge-Based Software Engineering (JCKBSE 2006)*, pages 167 – 175, July–August 2006.
6. Abon Chen, Hsin-Hsi Chen, and Polly Huang. Predicting social annotation by spreading activation. In *Proceedings of the 10th international conference on Asian digital libraries: looking back 10 years and forging new frontiers*, ICADL'07, pages 277–286, Berlin, Heidelberg, 2007. Springer-Verlag.
7. H. Chen and T. Ng. An Algorithmic Approach to Concept Exploration in a Large Knowledge Network (automatic thesaurus consultation): Symbolic Branch-and-Bound search vs. connectionist Hopfield net activation. *J. Am. Soc. Inf. Sci.*, 46(5):348–369, 1995.
8. Paul R. Cohen and Rick Kjeldsen. Information Retrieval by Constrained Spreading Activation in Semantic Networks. *Inf. Process. Manage.*, 23(4):255–268, 1987.
9. Allen M Collins and Elizabeth F Loftus. A spreading activation theory of semantic processing. *Psychological Review*, 82(6):407–428, 1975.
10. European Commission Communications. On the Digitisation and Online Accesibility of Cultural Material and Digital Preservation. Commission recommendation, European Commission, 2006.
11. F. Crestani. Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review*, (11):453–482, 1997.
12. H. Cui, J. Wen, J. Nie, and W. Ma. Query Expansion by Mining User Logs. *IEEE Transaction on Knowledge and Data Engineering*, 15(4):829–839, July 2003.
13. Alan J. Dix, Akrivi Katifori, Giorgos Lepouras, Costas Vassilakis, and Nadeem Shabir. Spreading Activation over Ontology-Based Resources: from Personal Context to Web Scale Reasoning. *Int. J. Semantic Computing*, 4(1):59–102, 2010.
14. Martin Fowler. *Refactoring: Improving the Design of Existing Code.* Addison-Wesley, Boston, MA, USA, 1999.
15. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns.* Addison-Wesley Professional, January 1995.
16. Qi Gao, Junwei Yan, and Min Liu. A Semantic Approach to Recommendation System Based on User Ontology and Spreading Activation Model. In *NPC '08: Proceedings of the 2008 IFIP International Conference on Network and Parallel Computing*, pages 488–492, Washington, DC, USA, 2008. IEEE Computer Society.
17. Jose Emilio Labra Gayo, Patricia Ordoñez de Pablos, and Juan M. Cueva Lovelle. Combining Collaborative Tagging and Ontologies in Image Retrieval Systems. 2007.
18. Jose Emilio Labra Gayo, Patricia Ordoñez de Pablos, and Juan Manuel Cueva Lovelle. WESONet: Applying semantic web technologies and collaborative tagging to multimedia web information systems. *Comput. Hum. Behav.*, 26:205–209, March 2010.
19. Fatih Gelgi, Srinivas Vadrevu, and Hasan Davulcu. Improving Web Data Annotations with Spreading Activation. In *WISE*, pages 95–106, 2005.

20. Stephan Gouws, G-J van Rooyen, and Herman A. Engelbrecht. Measuring Conceptual Similarity by Spreading Activation over Wikipedia's Hyperlink Structure. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 46–54, Beijing, China, August 2010. Coling 2010 Organizing Committee.
21. Maurice Grinberg, Vladimir Haltakov, and Hristo Stefanov. Approximate Spreading Activation for Efficient Knowledge Retrieval from Large Datasets. In *Proceeding of the 2011 conference on Neural Nets WIRN10: Proceedings of the 20th Italian Workshop on Neural Nets*, pages 326–333, Amsterdam, The Netherlands, The Netherlands, 2011. IOS Press.
22. Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*, volume 1. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool, 2011.
23. H.R.Turtle. *Inference Networks for Document Retrieval*. PhD thesis, University Illinois, Urbana, IL, USA., 1991.
24. Akrivi Katifori, Costas Vassilakis, and Alan J. Dix. Ontologies and the brain: Using spreading activation through ontologies to support personal interaction. *Cognitive Systems Research*, 11(1):25–41, 2010.
25. W. Liu, A. Weichselbraun, A. Scharl, and E. Chang. Semi-Automatic Ontology Extension Using Spreading Activation. *Universal Knowledge Management*, 0(1):50–58, 2005.
26. Jian-Yun Nie. Query Expansion and Query Translation as Logical Inference. *J. Am. Soc. Inf. Sci. Technol.*, 54(4):335–346, 2003.
27. Scott Everett Preece. *A Spreading Activation Network Model for Information Retrieval*. PhD thesis, University Illinois, Urbana, IL, USA., 1981.
28. Yonggang Qiu and Hans-Peter Frei. Concept-based query expansion. In *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 160–169, Pittsburgh, US, 1993.
29. Cristiano Rocha, Daniel Schwabe, and Marcus Poggi de Aragão. A Hybrid Approach for Searching in the Semantic Web. In *WWW*, pages 374–383, 2004.
30. Kinga Schumacher, Michael Sintek, and Leo Sauermann. Combining Metadata and Document Search with Spreading Activation for Semantic Desktop Search. In Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, editors, *Proc. of ESWC*, pages 569–583. Springer, June 2008.
31. Ján Suchal. On finding power method in spreading activation search. In Viliam Geffert, Juhani Karhumäki, Alberto Bertoni, Bart Preneel, Pavol Návrat, and Mária Bieliková, editors, *SOFSEM (2)*, pages 124–130. Safarik University, Kosice, Slovakia, 2008.
32. Proletina Todorova, Atanas Kiryakov, Damyan Ognyanoff, Ivan Peikov, Ruslan Velkov, and Zdravko Tashev. D2.4.1 Spreading Activation Components (v1). Deliverable or research project, LarKC FP7 project–215535, 2009.
33. Alexander Troussov, Mikhail Sogrin, John Judge, and Dmitri Botvich. Mining Socio-Semantic Networks Using Spreading Activation Technique. 2008.
34. By Michael Wolverton, Michael Wolverton, Michael Wolverton, Barbara Hayes-roth, and Barbara Hayes-roth. Retrieving Semantically Distant Analogies with Knowledge-Directed Spreading Activation. In *In Proceedings AAAI-94*, pages 56–61. AAAI Press, 1994.