Application of the Spreading Activation Technique for Recommending Concepts of well-known ontologies in Medical Systems

Jose María Álvarez[1], Pablo Abella[1], Weena Jimenez[1], and José Emilio Labra[1]

WESO RG, Universidad de Oviedo, Oviedo, Asturias, Spain,
{josem.alvarez,pablo.abella,weena.jimenez,jelabra}@weso.es,
WWW home page: http://www.weso.es

Abstract. The present paper introduces the ONTOSPREAD framework for the development, configuration, customization and execution of the Spreading Activation technique over graph-based structures, more specifically over 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 and Document Retrieval domains. ONTOSPREAD implements the double process of activation and spreading of concepts in ontologies 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. This technique provide a whole framework to ease the information access, common required feature in the exploitation of new and existing digital libraries. Finally 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 Spreading Activation technique (hereafter SA) introduced by [?], in the field of psycho linguistics and semantic priming, proposes 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. **??**, and it is relevant to the the digital libraries field in the scope of: 1) construction of hybrid semantic search engines and 2) ranking of information resources according to an input set of weighted resources. Thus this technique eases the information access providing a connectionist method to retrieve data like brain can do. Although SA is widely used, more specifically in recent years has been successfully applied to ontologies, a common and standard framework is missing and each third party interested in its application must to implement its own version [?] of SA.

Taking into account the new information realm and the leading features of putting together the SA technique and the Semantic Web and Linked Data initiatives, 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 legal document databases [?], e-procurement, e-tourism or e-health.

The proposed work aims to afford a framework for SA to ease the configuration, customization and execution over graph-based structures 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: 1) study and revision of the classical constrained SA; 2) study and definition of new restrictions for SA applied to RDF graphs and ontologies; 3) implementation of a whole and extensible framework (called ONTOSPREAD) to customize and perform the SA based; 4) outlining of a methodology to configure and refine the execution of SA and 5) an example of configuration and refinement applying SA over a well-known ontology, the Galen ontology.

2 Related Work

Since SA was introduced by [?] 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 [?,?]. Nevertheless the success of this technique is specially relevant to the fields of Document [?] and Information Retrieval [?]. It has been also demonstrated its application to extract correlations between query terms and documents analyzing user logs [?] and to retrieve resources amongst multiple systems [?] 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 [?,?] addressing the two important issues: 1) the selection and 2) the weighting of additional search terms and to measure conceptual similarity [?]. On the other hand, there are works [?] exploring the application of the SA on ontologies in order to create context inference models.The semi-automatically extension and refinement of ontologies [?] 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[?], 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 [?] services to gather meta-data [?] from the Web or to predict social annotation [?] and recommending systems based on the combination of ontolo-

gies and SA [?] are taken advantage of using SA technique. Also the semantic search [?] is a highlight area to apply SA following hybrid approaches [?,?] or user query expansion [?] combining metadata and user information.

Although this technique is widely accepted and applied to different fields open implementations[1], are missing. Moreover the Apache Mahout [2] 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.

3 ONTOSPREAD Framework

3.1 Background

In this section, the theoretical model of *SA* [?,?] is reviewed to illustrate the basic components and the operations performed by SA during their execution, specially the spreading of the activation from a node to their adjacent nodes. 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. SA is comprised of three stages: *Preadjustement* and *Postadjustement* that are usually in charge of performing some control strategy over the target semantic network and the set set of activated concepts and the *Spreading* stage in which concepts are activated in activation waves. 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

[1] Texai company (http://texai.org/) offers a proprietary implementation of SA.
[2] http://mahout.apache.org/

node to others. The activation rank I_i of a node n_i will change while algorithm iterates.

3.2 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 [?] 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): "highly connected" nodes 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 $\jmath$.

The aforementioned theoretical model is an excellent start point to design a framework 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 RDF graphs and ontologies.

Context of activation $\mathbb{D}_{com}$: the framework is able to manage some ontologies at the same time and concepts can be defined in different ontologies identified by a context URI (or namespace). The double process of activation and spreading will only be performed in the set of active contexts $\mathbb{D}_{com}$.

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 $c_l \in \Phi$[3] 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 $d_j = min\{d_{lj} : \forall n_l \in \Phi\}$:

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

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

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

3.3 Specification of the SA technique

The entry point to SA technique is the set of initial concepts that will generate a new set of the most relevant concepts. 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 the set of output concepts, where $n_i \approx c_i$ and $I_i \approx w_i$ (the relevance of the concept). The implementation of SA, see Algorithm **??**, 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 is the set of initial activated concepts, c_j^k is the spreading concept at the k-th 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.

[3] Set of initial concepts.

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

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

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:

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

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

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

Algorithm 1 *Pseudocode of Spreading Activation*

Require: $\Phi \neq \emptyset$
Ensure: $\mathcal{G} \neq \emptyset$
  $\mathcal{A} \leftarrow \Phi$
  $\mathcal{G} \leftarrow \emptyset$
  while $\mathcal{A} \neq \emptyset$ AND $card(\mathcal{G}) < \mathcal{G}_{\min}$ AND $N_k \geq N_{\min}$ **do**
    $n_k \leftarrow extract(\mathcal{A})$
    $\mathcal{G} \leftarrow \{n_k\} \cup \mathcal{G}$
    for all $n_i / w_{ki} > 0$ **do**
      $N_i \leftarrow N_i + w_{ki} N_k$
      $\mathcal{A} \leftarrow (\{n_i\} \cup \mathcal{A}) - \mathcal{G}$
    end for
  end while
  return $\mathcal{G}$

3.4 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 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*[4] *and they go through the node c_i and they only contain nodes belonging to*

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

$\mathcal{G}$. 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. **??**, 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.

3.5 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 that domain and considering it as a new stage of the ontology o graph modeling process. Since *SA* uses weights in relations to calculate the activation value of the concepts, different "patterns" have been identified to manage the direction of the spreading process: 1) Ascending seeks for the activation of concepts more generic than the current ("superclass"); 2) Descending seeks for the activation of concepts more specific than the current ("subclass"); 3) Nominal seeks for the activation of instances instead of concepts ("instance of") and 4) Crossing seeks for the activation of concepts and instances connected through a certain relation $\mathcal{R}$. 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.

3.6 Design and implementation of ONTOSPREAD

ONTOSPREAD framework[5], see Fig. **??** is addressed by an open and extensible design applying best practices on software design and development like design patterns and refactoring. The *Player* class handles the execution of the algorithm in a stepwise way. It 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*) that makes possible to serialize the current state and back to a previous one. On the other hand, the *SA* process comprises of three sub-processes (see Sect. **??**): *OntoSpreadPreAdjustment*; *OntoSpreadRun*; and *OntoSpreadPostAdjustment*. Moreover, the process carries on the information about the knowledge base using the DAO pattern thus the framework is independent from the modeling language of the semantic network (RDF-based vocabularies and OWL are now supported).

[5] http://code.google.com/p/ontospread/

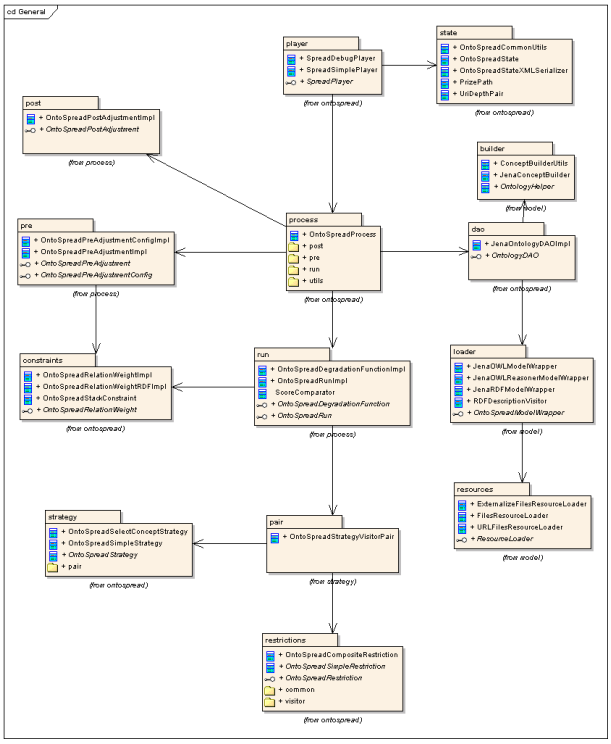

Fig. 1. ONTOSPREAD Overview Diagram.

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 (*OntoSpreadState*) 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). Moreover, 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 more 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. It 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 Evaluation of ONTOSPREAD

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 defined domain knowledge (concepts and relations) 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 their specification 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 well-known semantic network (ontology, etc.): concepts and relations.
2. Define a potential set of initial concepts (Φ) and their initial activation value (usually 1.0).
3. Specify the weights of the relations to that domain knowledge.
4. Combine the different restrictions provided by the framework.
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 to that domain knowledge.

To apply this methodology, the Galen Ontology [6] 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 weights of relations, the set of initial concepts, the set of restrictions, the degradation function and the extensions to reward nodes. The set of initial concepts (Φ) with an initial value 1.0 is: "#AdvancedBreastCancer" and "#NAMEDSymptom". 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 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.

After the execution of the different configurations, see Tab. **??**, some statistics have been extracted out of the results. The main differences between the tests lies in the number of activated nodes and their activation values due to the

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

Config & Stats/Test	T_1	T_2	T_3	T_4	T_5	T_6
Minimum activation value $N_{\min}$	1.0	1.0	1.0	1.0	1.0	1.0
Maximum number of spread concepts $\mathbb{M}$	50	50	10	10	50	50
Minimum number of spread concepts $\mathbb{M}_{\min}$	20	20	5	5	20	20
Output Degradation O_j	h_1	h_2	h_1	h_2	h_1	h_2
Reward (No,Yes)	N	N	N	N	Y	Y
Context of activation $\mathbb{D}_{com}$	DEFAULT					
Activated Nodes	62	79	15	15	62	79
Spread Nodes	20	20	5	5	20	20
Highest activation value	7.5	3.9896	1.5	1.90	7.5	3.9896
Deepest spread path	10	16	2	2	10	16
Concepts (name:value)	NAMED Symptom: 7.5, Primate: 2.28	Multi Cellular Eukaryota: 3.9896, Opis thokonts: 3.9885	NAMED Symptom: 1.5, Advanced Breast Cancer: 2.28	NAMED Symptom: 1.90, Advanced Breast Cancer: 1.90	NAMED Symptom: 7.5, Primate: 2.28	Multi Cellular Eukaryota: 3.9896, Opis thokonts: 3.9885
Time (msec.)	9	8	2	3	11	12

Table 1. Configuration and statistics of results after the execution and refinement of SA over the Galen ontology.

restrictions that guide the evolution of the algorithm through the graph and the structure of Galen ontology (23,141 concepts and 950 relations). It is also remarkable that the reward of paths, in this case, does not imply changes in the output set. This situation demonstrates that a depth knowledge of the semantic network is needed to take advantage of the SA extensions. Nevertheless the output of the algorithm helps us to establish a set of weighted resources that can be used to retrieve documents, make recommendations or search in large databases with enriched queries.

5 Conclussions and Future Work

This work provides a configurable and extensible framework 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 restrictions and weights of the relations, the selection of the degradation function and the use of reward functions. This framework minimizes these advantages with an extensible library 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 flexibility of the refinement methodology. An automatic learning algorithm to create SA configurations 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 measures 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. Finally a new version of the SA is being specified and developed following the Map/Reduce[7] programming model with the objective of getting a distributed version of this technique for processing large data sets.

Acknowledgements. ONTOSPREAD was initially developed in BOPA [?] project that is one of *Semantic Web Use Cases and Case Studies*[8] collected by W3C. Currently it is being applied to the process of searching public procurement notices in the "10ders Information Services"[9] 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"[10] and developed in cooperation with "Exis TI"[11] and WESO Research Group.

[7] http://labs.google.com/papers/mapreduce.html
[8] http://www.w3.org/2001/sw/sweo/public/UseCases/CTIC/
[9] http://rd.10ders.net
[10] http://gateway-scs.es/
[11] http://www.exis-ti.com/