

# ONTOSPREAD: Activation of Concepts in Ontologies through the Spreading Activation algorithm

Jose María Álvarez<sup>2</sup>, Diego Berrueta<sup>1</sup>, Luis Polo Paredes<sup>1</sup>, and José Emilio Labra Gayo<sup>2</sup>

<sup>1</sup> Fundación CTIC, Gijón, Asturias, Spain,  
{diego.berrueta,luis.polo}@fundacionctic.org,  
WWW home page: <http://www.fundacionctic.org>

<sup>2</sup> WESO RG, Universidad de Oviedo, {U01548,labra}@uniovi.es,  
WWW home page: <http://www.uniovi.es>

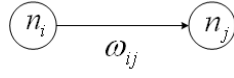
**Abstract.** The present article introduces the ONTOSPREAD API for the development, configuration, customization and execution of the Spreading Activation techniques in the field of the Semantic Web. These techniques have been used to the efficient exploration of knowledge bases based on semantic networks in Information or Document Retrieval areas. ONTOSPREAD enables the implementation of the Spreading Activation algorithm on RDF models and datasets, implicit graph structures. It implements the process of activation and spread of concepts in ontologies applying different restrictions like weight degradation according to the distance or the converging paths reward. The main application of Spreading Activation lies in two different areas: 1) construction of hybrid semantic search engines 2) ranking of resources according to an input set of weighted resources. Finally an evaluation methodology and an example using the Galen ontology are provided to validate the goodness and the capabilities of the proposed approach.

## 1 Introduction

### 1.1 Main Contributions

## 2 Background

In this section, the theoretical model of *SA* is reviewed to illustrate the basic components and the operations performed by the SA techniques during their execution. 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 take them as a semantic network in which SA can be applied. The processing 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 basic definitions made are presented:



**Fig. 1.** Graphical model of *Spreading Activation*

**Preadjustment:** 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 neighbour nodes.

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

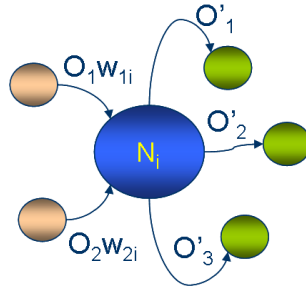
$$I_i = \sum_j O_j \omega_{ji} \quad (1)$$

$I_i$  is the total inputs of the node  $n_i$ ,  $O_j$  is the output of the node  $n_j$  connected to  $n_i$  and  $\omega_{ji}$  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.

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

$N_i$  is 1 if the node has been activated or 0 otherwise.  $j_i$ , 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 the 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

## 4 ONTOSPREAD

### 4.1 Constrained Spreading Activation

One of the main features of the SA techniques are their flexibility to fit to the resolution of different problems. From the configuration point of view some constraints presented in [?] have been customized improving the expected outcomes of the execution according to the domain problem. Following these constraints are defined:

**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 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 *iff* its activation value,  $I_i$ , is greater than a threshold activation constant  $j$ .

The abovementioned theoretical model in Sect.?? is a good start point to design an API for *SA* but from the domain expert point of view some requirements are missing (FIXME: ref) to provide a configurable set of techniques based on *SA*. 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 carried out in the context  $\mathbb{D}_{com}$ .

**Definition 1.** *Let  $\mathbb{D}_{com}$  an active domain, if a concept  $c_i$  is activated or 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  $M$  :** the process of activation and spreading will be executed, at the most, until  $M$  concepts had been spread.

**Minimum number of spread concepts  $M_{min}$  :** the process of activation and spreading will be executed, at least,  $M_{min}$  concepts had been spread.

**Time of activation  $t$ :** the process of activation and spreading will be executed, 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.

$$O_j = h(I_j) \quad (3)$$

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

$$O_j = h_0(I_j) = I_j \quad (4)$$

2. Customization using **distance**:  $h_1$  calculates the level activation of the concept  $c_j$  according to the distance from the initial concept  $c_l \in \Phi^3$  that has activated it. The activation value have to decrease if the distance from  $\Phi$  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 the concepts (decreasing their rank) far from the “activation core” and rewards closed concepts. Thus, let  $d_j$ , where  $d_j = \min\{d_{lj} : \forall n_l \in \Phi\}$ :

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

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

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

## 4.2 Design of Spreading Activation

The entry point to *SA* techniques is the set of initial concepts ( $Q_{sem}$ ) that will generate a new set of the most relevant concepts ( $Q'_{sem}$ ). Ontologies are a graph where each node  $n_i$  represents a concept  $c_i$  and the edge  $\omega_{ji}$  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 build  $Q'_{sem}$ , where  $n_i \approx c_i$  and  $I_i \approx w_i$  (the relevance of the concept).

The implementation of *SA*, see Algorithm.4.2, 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)  $\Phi^4$  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).

$$\mathcal{A}^0 = \Phi \quad (7)$$

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

---

<sup>3</sup> Set of initial concepts.

<sup>4</sup>  $\Phi \equiv Q_{sem}$ .

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

$$\mathcal{G}^0 = \emptyset \quad (9)$$

$$\mathcal{G}^k = \mathcal{G}^{k-1} \cup \{c_j^k\} \quad (10)$$

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:

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

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

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

---

**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**  $\text{card}(\mathcal{G}) < \mathcal{G}_{\min}$  **AND**  $N_k \geq N_{\min}$  **do**

$n_k \leftarrow \text{extract}(\mathcal{A})$

$\mathcal{G} \leftarrow \{n_k\} \cup \mathcal{G}$

**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**  $\mathcal{G}$

---

### 4.3 Improving Spreading Activation

Some improvements in the calculus of the activation value of a concept have been introduced in order to get FIXME. If some paths of activation converge to the same node and the source nodes are different then this node should be important 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  $\Phi^5$  and they go through the node  $c_i$  and they only contain nodes belonging to

---

<sup>5</sup> The reward is not applied to nodes in  $\Phi$ .

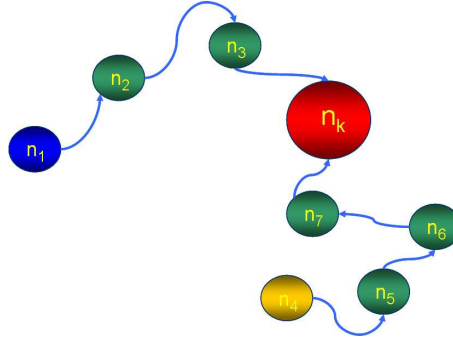
$\mathcal{G}$ . This improvement assigns a new value to the activation value of each node  $c_i$  indicated by  $I_i^*$  and it is calculated by means of the function  $g$ :

$$I_i^* = g(I_i, p_i) \quad (13)$$

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 behaviour of *SA* algorithm is not broken.

$$g(x, y) = x(\log(y + 1) + 1) \quad (14)$$

$x$  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.



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

#### 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 performed by a domain expert taking into account the specific issues of this domain and considering as a new stage of the ontology o graph modelling process.

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 are the key point to customize *SA* techniques.

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

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

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”
Cross	It seeks for the activation of concepts and instances connected through a certain relation $\mathcal{R}$ .	$\mathcal{R}$

**Table 1.** Patterns of direction control in *SA*.

#### 4.5 Refining Spreading Activation

#### 4.6 Implementation of ONTOSPREAD API

#### 4.7 ONTOSPREAD tools

#### 4.8 Use Cases and Scenarios

### 5 Evaluation of ONTOSPREAD API

#### 5.1 ONTOSPREAD API in Action

## 6 Conclusions

### 6.1 Future Work