

# 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>

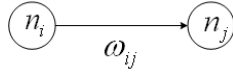
**Abstract.** The present article introduces the ONTOSPREAD API for the development, configuration, customization and execution of the Spreading Activation techniques in the field of the Semantic Web. These techniques have been used to the efficient exploration of knowledge bases based on semantic networks 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 node in spreading activates its neighbour nodes.

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

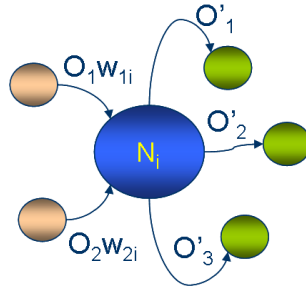
$$I_i = \sum_j O_j \omega_{ji} \quad (1)$$

$I_i$  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  $\gamma$ .

### 4.2 Design of Spreading Activation

### 4.3 Improving Spreading Activation

### 4.4 Refining Spreading Activation

### 4.5 Implementation of ONTOSPREAD API

### 4.6 ONTOSPREAD tools

### 4.7 Use Cases and Scenarios

## 5 Evaluation of ONTOSPREAD API

### 5.1 ONTOPSREAD API in Action

## 6 Conclusions

### 6.1 Future Work