How to Compute the References Emergencies in a Hyper-encyclopedia

BOGDAN PATRUT and IOANA PANDELE
Department of Mathematics and Computer Science
University of Bacau
600114 Bacau, Spiru Haret, 8
ROMANIA

Abstract: In this paper we present an agent oriented implementation of the transfer of the references emergencies in a hyper-encyclopedia. First of all, we formally define a hyper-encyclopedia as a reunion of encyclopedias which belong to some network nodes. Furthermore we present a model of the hyper-encyclopedia based on oriented graphs. Finally, we propose a distributed algorithm which describes the functionality of some agents within their nodes. Such agents transmit tasks from one node to another, regarding the definition of new (encyclopedias) articles. These tasks are hierarchically structured by an evaluation function which describes both the importance and the emergency of one task. We also present some efficient modalities of calculus.

Key-Words: Encyclopedia, rank algorithms, graphs, software agents

1 Introduction

Wikipedia is an example of a high success distributed encyclopedia, which can compete with well-known encyclopedias like Britannica or Encarta. By one hand, Wikipedia does not have a hierarchical system based on references (mutual citations), on the other hand, Google uses PageRank algorithm in order to hierarchize the web pages. PageRank uses simple mathematical functions applied to the web pages super-graph. There are different other formulas [3,4,5] for page ranking, each of them with practical advantages and disadvantages. We will suggest such a system to hierachize the articles in a hyper- encyclopedia. As a result we will define a measure of the importance of existing articles and a measure of the emergency in defining new articles, based on the citations of the existing ones.

2 The problem

2.1 Developing a hyper-enciclopedya

We consider a computer network. We suppose that in each node we have some human users (students) who want to develop a hyper- encyclopedia. A hyper-encyclopedia is a distributed encyclopedia. The articles are defined in different network nodes and each article could have (within his definition) links, both to articles in the same node and to articles in other nodes.

A hyper-encyclopedia is practically a network of encyclopedias, in each node there exists one encyclopedia (figure1): $E=E_1 \cup E_2 \cup E_3 \cup \ldots \cup E_n$, where E_i is the encyclopedia developed in node i, $i=\overline{1,n}$.

Each user i has to develop a certain theme or encyclopedic document Dom_i. It is possible and permitted for most users to introduce data and, thus, to define the same entry in their encyclopedias, but with different definitions and in different nodes.

One encyclopedia E_i is a collection of articles, by the form $E_i = (Dom_i, \{A_{ij}\}\ j=\overline{1,M_i})$, where M_i is the number of articles from encyclopedia E_i, $Q_i=|E_i|$. An entry-article A_{ij} in E_i encyclopedia has the form $A_{ij}=(T_{ij},D_{ij},LS_{ij})$, where T_{ij} is the article title, and D_{ij} is its definition. The article definition includes different links from the LS_{ij} set to another articles of hyper/encyclopedia. One link could be to one article in the same encyclopedia, or to another one, from another encyclopedia.

One link $L_{ijk} \in LS_{ij}$ has the form $L_{ijk}=(u,v)$ which means that L_{ijk} is a references to the article u from encyclopedia numbered with v. Thus, the encyclopedia of user i, is: $E_i = (Dom_i,\{A_{ij}\})$, where A_{ij} defines the j-article from encyclopedia numbered with i.

The users have to develop the encyclopedias. The development implies writing articles, which means

ISBN: 978-960-6766-91-6

ISSN 1790-2769

writing their titles, definitions, and editing the links within their definitions to other articles which exist or not.

Each user will introduce the articles of his own encyclopedia. When he introduces one particular article A in his own encyclopedia, the student has to write the title, then the article's definition, and to make references to other articles related to A. In this way, the student has to create a link between A and other articles from his own encyclopedia or from the others. The user introduces the title himself; for the definition he could use books, journals, web resources etc. In the mean time, he could find inspiration from the definitions (of the same article) that his colleagues could give to him. In this way, he will access the encyclopedias of his colleagues and he will search for article A. He will take the definitions of A, given by his colleagues and he will create his own.

In order to realize the references of this article to others articles, these articles must exist, if not they must be created afterwards. If these articles do not yet exist, they became a task for a particular user. Every user will have, in this way, more tasks, which means the introduction in his encyclopedia of a definition for one term or another.

Initially, the user will think at some articles from his encyclopedia's domain, to whom A bears on. He will search for these articles in his encyclopedia and, if they exist he will create the links. If not, he will make the links, but he will note in his task scheduler to append this article in his encyclopedia.

After that, he will look to create links between his article and articles related to this one from other encyclopedias. He will search at his colleagues those articles related to A, or those which are linked to A and he will create the proper links. If every student has a well defined domain to fulfill then the links between A and the articles from other encyclopedias will be made according with the domain (or domains) where A belongs.

2.2 An example of hyper-encyclopedia

Assume that some students (users) develop a hyper-encyclopedia about computer sciences. In figure 1, the assignment of student number 1 is programming languages. Student number 2 has the assignment peripheral equipment and student number 3 has to write about algorithms and data structures.

Let us suppose that student number 1 has introduced already the articles called Pascal and C, and now he has to introduce the article about Basic. Suppose that student 2 and 3 have introduced their articles as we can see in figure 3. At the moment when student 1 defines the article "Basic" from his encyclopedia, he could make the following types of links:

1. links to already defined articles, from his encyclopedia, like Pascal or C – in this case he only creates the links and they became functional;
2. links to other articles that do not yet exist in his encyclopedia, but they should be, for example Fortran or Java. In this case, he will create the link (which is not functional for the moment) and he will append in his tasks list that he must introduce these articles;
3. links to articles which already exist in other encyclopedias, like "Monitor" or "Keyboard" from encyclopedia 2 or "Input/output operations" in Basic); in this case the links are created and they are functional;
4. links to articles that do not exist in his encyclopedia, but they should be in other encyclopedias (they are in the task lists of those students); in this case the links are created, although they are no yet functional.

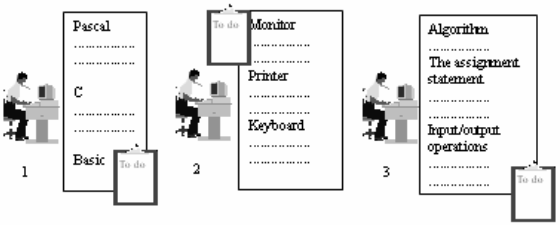

Fig.1. An example with three students. The articles already introduced and the tasks lists

From this example we notice that every student has to have a list with the articles to append in his own encyclopedia. It is more suitable that each student can append articles in his colleague's tasks lists if he considers that his colleagues should introduce some particular articles and they did not (or they did not suggest it). For example student number 1 could consider that in encyclopedia 3 there should exist a definition for the "repetitive control structure", but maybe student number 3 has not introduced such a notion and does not intend to. In this case, student 1 appends this article to student's 3 tasks list.

3 The solution

3.1 An oriented graph structure for the hyper- encyclopedia's articles

If we will consider that the articles (terms) from the whole encyclopedia and their related links, form an oriented graph $G = (NG, MG)$, then each network

ISBN: 978-960-6766-91-6

ISSN 1790-2769

node has in fact, a part from *G* nodes and also a part from arc, corresponding to the internal links.

Let *X* be *X = (NX, MX)*, and *Y = (NY, MY)* two subgraphs of this kind for *G*, which correspond to some encyclopedias within the network nodes. For an oriented subgraph *X*, let *NX* be the set of *X* vertices. The vertices are words defined in the corresponding node of *X*. Thus, the graph *X = (NX, MX)* corresponds to an encyclopedia *i*, $i=\overline{1,n}$, where *NX = {Aij}* $j=\overline{1,Q_i}$ and *MX = {L∈ LSij; L = (u, i),* $\exists\, k=\overline{1,Q_i}$ *: A_{ik}=u}*. We also define by D_v^X the text of the definition of the term *v∈NX* in *X*. The length of one definition is given by its number of words. Let *length(D_v^X)* be the length of definition of *v* in *X*. We will consider a certain term *v* from *NX*. If in the definition of *v*, a reference to the term *u* occurs (no matter if *u* belongs to *NX* or to *NG-NX*), we will define by *position(u,D_v^X)* the occurrence position (the distance in words from the beginning of the definition). We will connect to the oriented graph *G* a cost matrix *A=(a_{vu}), uv∈N*, with (34.1):

$$a_{vu}=\frac{position(u,D_v^X)}{length(D_v^X)} \qquad (3.1)$$

3.2 An estimation function for determining the emergency in defining an article within a node

Based on the notations above, we define the importance of a term *u∈NX*, in the subgraph *X*, by (3.2): *importance (u, X) =*

$$\left(\sum_{v\in G-X}a_{vu}+\sum_{v\in X}a_{uv}\right)-\left(\sum_{v\in X}a_{vu}+\sum_{v\in G-X}a_{uv}\right) \qquad (3.2)$$

The importance of one term within a definition is a measure of the position where this term should appear (in the connected encyclopedia) and it is related to terms already defined.

For non-existing terms, but with existing references (in their own encyclopedia, or in others), we define a node emergency. The emergency of node *u* referred in *X*, and which should be defined in *Y*, is given in formula (3.3).

emergency (u, Y, X) =

$$\left|\{v\in N_X \mid a_{vu}>0\}\right|\times\sum_{v\in N_X}a_{vu} \qquad (3.3)$$

In the node *Y*, for a term *u*, we can calculate the emergency in defining this term by summation of the emergencies received from each node *X* from graph *G*.

total-emergency (u, Y) =

$$\sum_{X\subset G}emergency(u,Y,X) \qquad (3.4)$$

3.3 The agent for transferring the emergencies of the articles from one to another

An agent is a program that is capable of acting by him in order to achieve its goals. It runs without human intervention, proving pro- activity, reactivity to environment and communication skills [1].

The transfer agent (*ATR*) associated with node *X* will work like this:

- it will determine what new articles (new tasks) should be asked to the other nodes, in connection with the underlying domains (according to the user specifications from X);
- it will receive from the network new articles (new tasks) that it has to define and append these to NX;
- it will calculate the emergencies of the terms referred in NX and send them to nodes that have to define these terms;
- it will receive from the network the emergencies of the terms witch have to be defined in NX;
- it will rank decreasingly the list of nodes that must be defined in NX, according to the values of total emergencies and show this list to the user.

Hence, the algorithm for agent *ATR* from node *X* will be:

```
agent ATR(X)
for each v∈N_X
   for each Y from G
     for each u∈N_Y
          a_vu = position(u, D_v^X) / length(D_v^X)
     end-for
     New_Tasks_for_Y=∅
     for each link y from D_v^X
      receive Dom(y) from user(X)
      if Dom(y)=Y then
        New_Tasks_Y=New_Tasks_ Y ∪ y
      end-if
      send New_Tasks_Y to ATR(Y)
     end-for
  end-for
end-for
```

ISBN: 978-960-6766-91-6

ISSN 1790-2769

```
L = ∅
for each Y from G
  receive New_Tasks_X from ATR(Y)
 for each task from New_Tasks_X
   N_X = N_X ∪ New_Tasks_X
   receive N_Y from ATR(Y)
   for each u∈N_Y
     k=0
     for each v∈N_X
        if a_vu>0 then k=k+1
     end-for
     e=0
     for each v∈N_X
        e=e+k*a_vn
     end-for
     emergency(u,Y,X) = e
     send emergency(u,Y,X) to ATR(Y)
   end-for
   for each v∈N_X
     receive emergency(v,X,Y)
             from ATR(Y)
     L = L ∪ {emergency(v,X,Y)}
   end-for
   send N_X to ATR(Y)
 end-for
 total-emergency(u,Y) =
```

$$\sum_{X \subset G} emergency(u, Y, X)$$

```
 sort L descending,
       criteria total-emergency
 send L to User(X)
end agent
```

This algorithm is based on messages exchange, it is correct and it functions in a network of nodes. It was implemented in a multi-agent system for assistance in natural language (MAgeLan). MAgeLan is developed in the Laboratory of Artificial Intelligence and Cognitive Science (LAICOS), at the University of Bacău, Romania. We used MAgeLan for the Rocarta encyclopedia [2] with 10 nodes (10 different domains) and a total of 2295 articles. The ATR agents hyerarchized the articles and they used the process of developing the encyclopedia.

4 Conclusion and future work

In this paper we found a measure to define the importance of the existing articles from an hiper-encyclopedia. We also defined the emergency of the new cited articles. The hiper-encyclopedia is a reunion of local encyclopedias, belonging to the nodes of a network. In the future, we want to modify the algorithm, such that it would function when some nodes will fail.

References:

[1] G. Weiss (ed.), *Multiagent systems: A modern Approach to Distributed Artificial Intelligence*, MIT Press, Cambridge, MA

[2] Bogdan Pătruț, *Enciclopedia multimedia Rocarta şi agenții inteligenți*, Proceedings of Conferinței Universitatea Virtuală în România, Academia de Studii Economice, Bucureşti, 2003;

[3] *** - *IBM + The CLEVER search engine http://www.almaden.ibm.com/projects/clever.shtml*

[4] Kleinberg; J. M, *Method and system for identifying authoritative information resources in an environment with content-based links between information resources,* http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=6112202.PN.&OS=PN/6112202&RS=PN/6112202

[5] ***, *TrustRank algorithm,* http://pagerank.suchmaschinen-doktor.de/trustrank.html

[6] ***, *Page Rank algorithm*, http://pr.efactory.de/e-pagerank-algorithm.shtml

ISBN: 978-960-6766-91-6

ISSN 1790-2769