Three-Dimensional Representation of Conceptual Fuzzy Relations

Jose A. Olivas and Samuel Rios

Dept. of Information Technologies and Systems
University of Castilla-La Mancha
Paseo de la Universidad 4,
13071, Ciudad Real, Spain
Joseangel.olivas@uclm.es,
i92riros@uco.es

Abstract. In this work, T-DiCoR is presented (**T**hree **Di**mensional **Co**nceptual **R**epresentation) as a tool for representing the fuzzy relations among the most representative concepts of a domain. Using this tool in a Metasearcher, the user may observe what other concepts are related to the searched concept, and what the connection forces are (fuzzy relations between concepts). This knowledge can be useful for making new queries with words conceptually related in a specific domain with the original ones.

1 Introduction

Nowadays, the task of recovering information of big data sources, especially the Web, has great relevancy to many users of different areas and of the whole world. The search engines play a determinant role in this task, but increasingly, the users need help to centre the search on their aims.

In this work a tool is presented that can be useful to centre the search, since it allows showing the relations among the most relevant concepts of a certain domain. This allows the user to refine his queries, verifying what concepts are the most relevant in the domain and with which ones and how they are related.

The presented tool T-DiCoR (Three Dimensional Conceptual Representation) shows the user a three-dimensional form, a graph with the form of molecule, where the nodes are the most relevant concepts of a domain and the edges show the forces (fuzzy) that join them. These edges will be represented in different colours and thicknesses according to the intensity of the relation between the concepts. The user can change the view of the graph using the mouse.

It is very important to know how to get the input matrix that represents the fuzzy relations between the concepts. It is done with a fuzzy aggregation of different values from different sources (as is shown in point 2). In point 3 the T-DiCoR tool and the algorithms used are described. A complete example is explained in section 4, and the paper finishes with some conclusions and future trends.

H. Legind Larsen et al. (Eds.): FQAS 2006, LNAI 4027, pp. 681 – 690, 2006.

2 The Input Matrix

The input matrix contains the fuzzy relations among the 20 most important concepts of a specific domain. The weight values came from the aggregation of several different sources:

2.1 Contextual, Linguistic and Ontology Relations (FIS-CRM vectors)

FIS-CRM [1] is a model for representing the concepts contained in any kind of document. It can be considered an extension of the vector space model (VSM). Its main characteristic is that it is fed on the information stored in a fuzzy synonym dictionary and various fuzzy thematic ontologies. The dictionary stores the synonymy degree between every pair of recognized synonyms. Each ontology stores the generality degree between every word and its more general words. The way of calculating this value is the one proposed by Widyantoro & Yen in 2001 [2].

The key of this model is first to construct the base vectors of the documents considering the number of occurrences of the terms (what we call VSM vectors) and afterwards to readjust the vector weights in order to represent concept occurrences, using for this purpose the information stored in the dictionary and the ontologies.

The readjustment process involves sharing the occurrences of a concept among the synonyms which converge in the concept and give a weight to the words that represent a more general concept than the ones contained.

2.2 Causal Relations

To detect the causal relationships that exist in a collection of documents, a starting point could be to detect conditional phrases. Nevertheless, this is not an easy task. Descartes could not have possibly imagined that to propose his famous phrase "I think, therefore I am", would have given birth to so many conjectures and interpretations for centuries after. In reality, what did he want to say, "First I think and after I am a person, or As I am capable of thought, I am a person".

To sum up, even on this occasion the intention of Descartes seems clear when he expressed his maxim, it is not easy to interpret and format the information expressed in natural language, especially when it involves complex sentences with complicated turns.

With the aim of detecting conditional phrases, we have developed a basic system [3] of detecting structures and a classification of sentences that allows us to locate, in terms of basic components (verb tenses, adverbs, linguistic turns, etc.), certain causal forms.

To make the grammatical analysis, we have observed on the one hand, that we can separate certain causal relationships based on the verb form used, while on the other hand we can separate others based on the adverbs used in the sentences. Both analyses give rise to some causal rules that we will use afterwards to make an automatic extraction of knowledge. In the same way, every structure is subdivided into two structures which correspond to the antecedent and consequence of the causal relationship, and a parameter that measures the degree of certainty, conjecture, or compliance of the said causal relationship. In other words, it is not the same to form a sentence such as: “If I win the lottery, I will buy a car”, in which there is no doubt that if the antecedent comes true the consequence will come true, as to form the sentence “If we had bought a ticket in Segovia, we could have won the lottery” which leaves many more doubts and conjectures, in which you cannot be sure that the completion of the antecedent guarantees the consequences.

2.3 Aggregation of Fuzzy Values

The aggregation for constructing the input matrix is still an open problem. Nowadays it is done using standard OWA operators [4] or more sophisticated ones, such as the ones presented by Castro and Trillas [5], for example:
Arithmetic Weighted Generalized Mean: $m(A, B)(x) = \sup_a A(a) \wedge B((x-(1- p)a)/p)$, $p\in [0, 1]$
Geometric Weighted Generalized Mean: $m(A, B)(x) = \sup_a A(a) \wedge B((x/a^{(1-p)})^{1-p}$, $p\in [0, 1]$, or
Fuzzy Weighted Generalized Mean: $m(A, B)(x) = p\, A(x) + (1-p)\, B(x)$, $p\in [0, 1]$.

3 T-DiCoR

T-DiCoR (Three Dimensional Conceptual Representation) is designed to show the user a three-dimensional graph with the form of molecule, where the nodes are the most relevant concepts of a domain (up to a maximum of 20, for visualization reasons) and the edges are the forces (fuzzy) that join them (a concept will take 7 relations as a maximum, for the same reason). These edges will be represented in different colours (from the coldest up to the hottest) and thickness according to the intensity of the relation between two concepts. The user can rotate the graph using the mouse. The user can also choose one or more concepts to see them separately (with their relations).

3.1 Used Technology

The tool has been developed with Microsoft Visual Studio 2005 (V. 8.0.50727.42) in C# language.

3.2 Transformation: Fuzzy Conceptual Matrix to Fuzzy Conceptual Graph

The Conceptual Matrix is characterized for being square and triangular, in which every row represents the conceptual relations that every term has with the others. Due to this, the matrix only contains floating point numerical values, there must exist another vector that contains the list of ordered terms so that the relative position with the above mentioned vector coincides with the position inside the matrix.

The matrix will transform into a graph in which every node will represent a term and every edge will represent the existing relation between two concepts. The steps that the algorithm will follow are the following:

Being that:

F: the data file that contains the terms and the relations.
t: a term.
M: the matrix.
f: a row of the matrix.
c: a column of the matrix.
L: list of terms.
G (V, E): the graph where V are the nodes and E the edges.

Transformation Matrix-Graph (Matrix M, Graph G):

1.-To read the data file and extract the list L of terms and the matrix M of information.
2.-For each t ∈ L
  2.1. To insert a node v in V
3.-For each f ∈ M
  3.1.-For each c ∈ M when c> f
    3.1.1. To create an edge from the term [row] up to the term [column]

3.3 Representation of the Fuzzy Conceptual Graph

3D Model

The graph was designed in a 3D model due to the fact that this model facilitates the visualization of a graph that will be complex enough.

Chemical Molecule

The 3D representation is based on a model used in another tool that visualizes chemical molecules, adapting it to our needs [6]. The particularity of this visor is that it allowed rotating the molecule on itself using the mouse.

Really, the representation is in 2 dimensions and the nodes are shown in the position of their coordinates XY, but to simulate a 3 dimensional space, when we rotate the graph with a single drag-and-drop with the mouse left button, the tool uses a mathematical model that in addition to re-calculate the positions XY, it also calculate the Z coordinate of all the nodes. When it re-draws the nodes in their new positions, the edges are easy to draw.

The mathematical model that simulates the graph rotation is based on the calculation of the positions by matrices, but it could be summarized as follows:

1. First, we have to calculate the rotation angles for the two dimensions of the representation:

$$X_{angle} = \left(prev_y - y\right)\left(\frac{360}{width}\right) \tag{1}$$

$$Y_{angle} = \left(x - prev_x\right)\left(\frac{360}{height}\right) \tag{2}$$

Being that:

$prev_x/prev_y$: Previous mouse 'x' and 'y' coordinates, that is, the coordinates that we obtain when the user clicks on the graph drawing.

x/y: Actual mouse 'x' and 'y' coordinates.

width/height: Dimensions of the representation zone.

2. Then, when we have obtained the angles, the next step is to calculate the new positions of all the graph nodes as follows:
   a. We could transform the 'y' and 'z' coordinates of the nodes position with the 'Xangle':

$$ct = \cos(Xangle)$$
$$st = \sin(Xangle) \tag{3}$$

$$n_y = n_y \cdot ct + n_z \cdot st$$
$$n_z = n_z \cdot ct - n_y \cdot st$$

   The node position vector is:

$$N = (n_x, n_y, n_z)$$

   b. And the 'Yangle' could transform the 'x' and 'z' coordinates:

$$ct = \cos(Yangle)$$
$$st = \sin(Yangle) \tag{4}$$

$$n_x = n_x \cdot ct + n_z \cdot st$$
$$n_z = n_z \cdot ct - n_x \cdot st$$

   The node position vector is:

$$N = (n_x, n_y, n_z)$$

In conclusion, each node final position will be calculate using the (3) and (4) formulas.

Positioning Nodes. Fruchterman – Reingold Algorithm

Once the graph is generated, the only thing that we must still do is to redistribute the nodes of the same one so that its representation is the clearest possible (that is to say, avoiding crossings of edges and that the nodes stay closer to one another). For this, the algorithm of Fruchterman - Reingold [7] is used, it is a model based on the tracing of graphs directed by forces.

The method that it uses is based on the Eades' *spring embedder* [7], which can be briefly explained by saying that the nodes of the graph are like positive charges that are repelled, and the edges that join them are like springs that exercise a force of attraction between them (Hooke's physical law).

The algorithm of Fruchterman - Reingold is implemented as follows:

Notation:

$$G = (V, E)$$

$$p_v = (x_v, y_v, z_v)$$

$$\overrightarrow{p_u p_v} = \frac{p_u - p_v}{\|p_u - p_v\|_2} \tag{5}$$

$$d_{uv} = \left\|\overrightarrow{p_u - p_v}\right\| = \|p_u - p_v\|_2 \tag{6}$$

Repulsive force between nodes:

$$k = C\sqrt{\frac{a}{|V|}} \tag{7}$$

$$f_r(u,v) = \frac{k^2}{d_{uv}} \overrightarrow{p_v p_u} \tag{8}$$

Where:

C: experimental constant
a: area of the surface where the graph is going to be represented
|V |: number of nodes of the graph

Attractive force of the edge:

$$f_a(u,v) = \frac{d_{uv}^{\ 2}}{k} \overrightarrow{p_v p_u} \tag{9}$$

Total force on a node:

$$f(u,v) = \sum_{(u,v)\in E} f_a(u,v) + \sum_{v\in V, u\neq v} f_r(u,v) \tag{10}$$

Algorithm:

FR (Graph G)
1. Assign an initial temperature (t = t_0)
2. Randomly assign a position to every node v of G
3. Repeat M times:
   3.1.-For every $v \in V$
      3.1.1. Calculate $f(v)$ (according to the previous formula)
   3.2.-For each p_v
      3.2.1. $p_v = p_v + \frac{f(v)}{\|f(v)\|} \min(\|f(v)\|, t)$ (11)
      3.2.2. Control that p_v does not stay out of the area of drawing
      3.2.3. t = cool (t)

Painting the Edges According to Their Weights

The last important aspect to consider is the weight of the relations between terms. The idea of painting the edges with a specific thickness and colour is used to represent the degree of relation that the connected terms have. So, a small weight will be represented by a thin line and a cold colour such as the green. Nevertheless, a strong relation will be represented by a thick red line. The intermediate relations will be represented with thicknesses and colours spread between both ends.

4 An Example

An example of the tool is presented. The domain used is "My home". A set of documents of the field were retrieved, and the 20 most relevant concepts were: *Home, place, domicile, abode, habitation, dwelling house, house, household, legal residence, base, property, space, family, ménage, position, situation, placement, stance, emplacement* and *location*. Then, the fuzzy relations matrix is the one that it is shown in table 1.

Table 1. Input matrix for the domain "My home"

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1 home		.66	.83	.44	.36	.78	.86	.88												
2 place			.53		.32		.68				.94	.91				.76				
3 domicile									.96	.52				.46				.38		
4 abode					.32		.41		.95	.65		.58								
5 habitation											.55									
6 dwelling house																				
7 house											.65	.62		.54						
8 household																				
9 legal residence											.78									.62
10 base															.74	.71	.56		.92	
11 property																				
12 space														.89				.64		.51
13 family														.79						
14 menage																.94				
15 position																			.74	.75
16 situation																	.81			.88
17 placement																			.99	
18 stance																				
19 emplacement																				.92
20 location																				

Table 1 shows the fuzzy relations between concepts, extracted from the aggregation of the fuzzy values of its linguistic, contextual, ontological and causal relations among the concepts.

Figure1 shows the tool interface and the graph generated to visualize in 3D all the relations of the input matrix. It can be observed that 20 concepts could be the limit for a correct visualization of all concepts and relations.

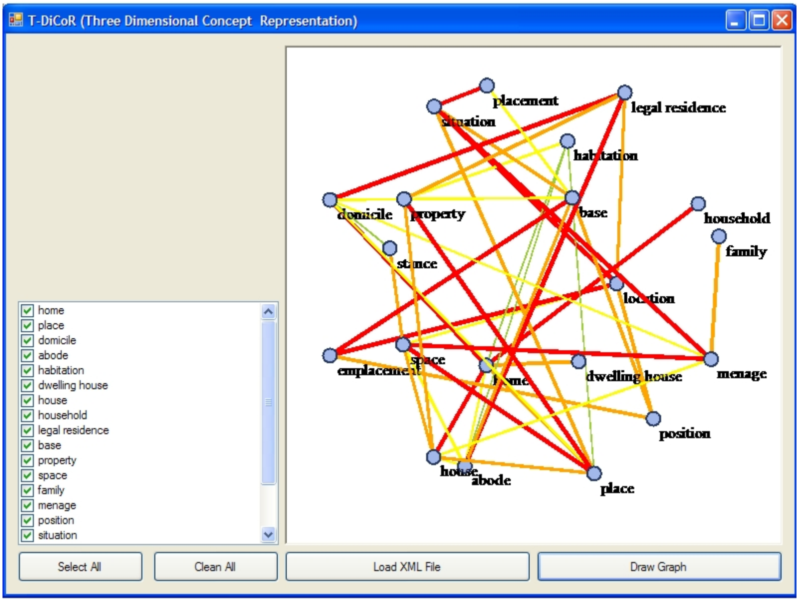

Fig. 1. Graph among all the concepts

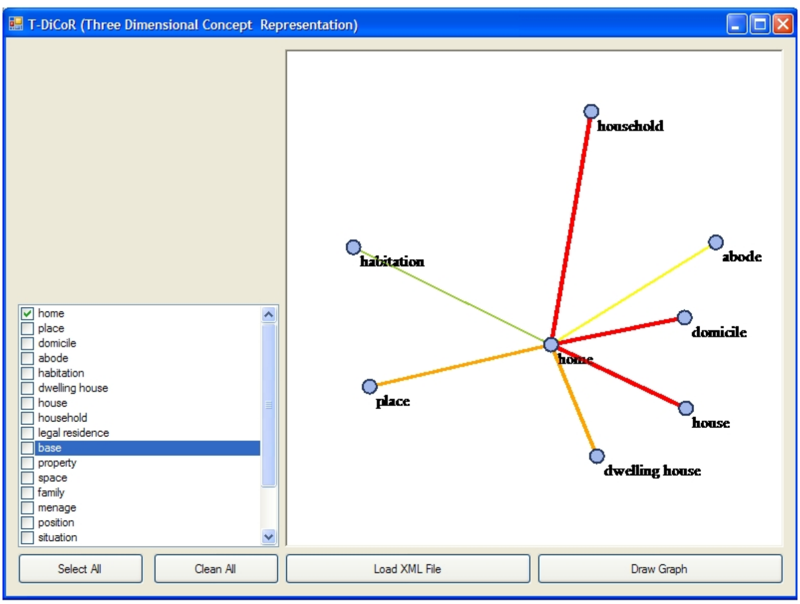

Fig. 2. Graph related to the concept “home”

Figures 2 and 3 depict the detailed relations of the concept “home”, the concepts “home” and “base”, and the concepts “home”, “base” and “position” (figure 3).

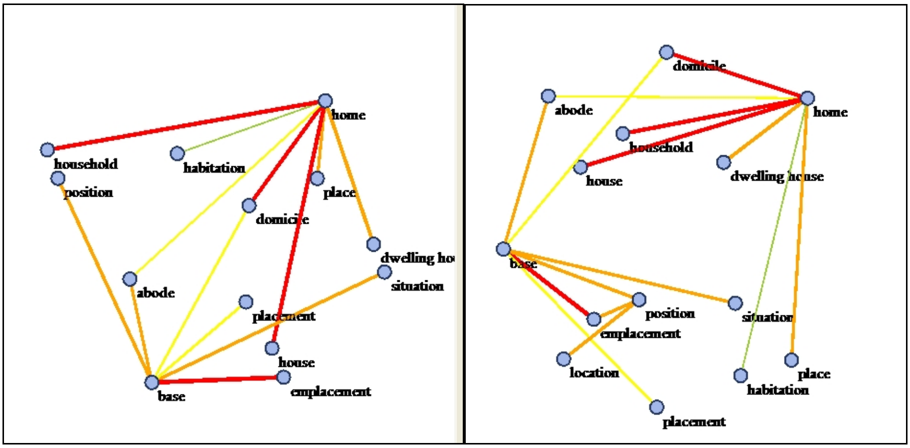

Fig. 3. Graphs related to the relations between “home” and “base” and to the relations between “home”, “base” and “position”

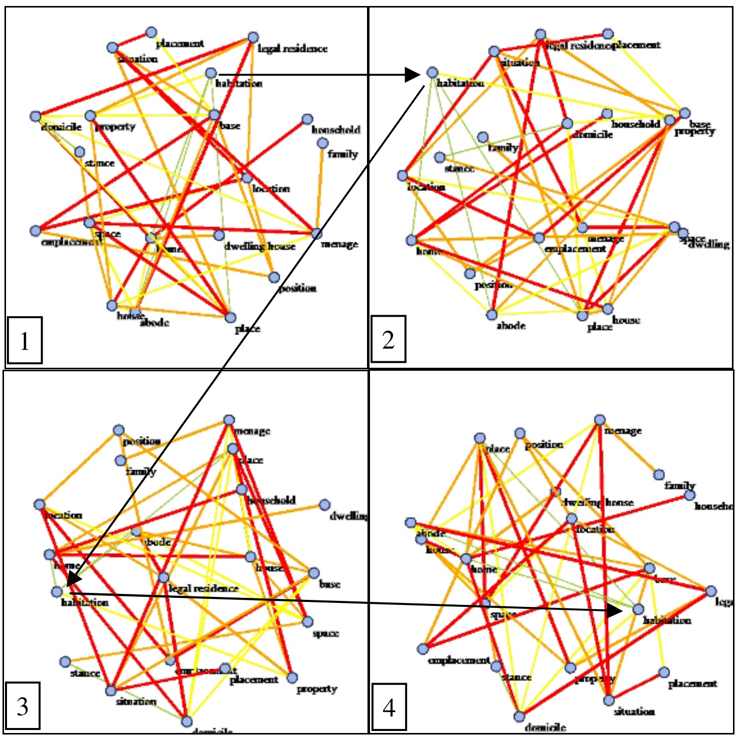

Fig. 4. Example of a 3D movement

In figure 4, a mouse 3D movement done by the user can be observed. Since it is difficult to represent 3D movements in a static 2D figure, the movements in figure 4 can be followed focusing our point of view in the concept “habitation”. First, there is a rotation to the left on the X axe (1→2) followed by a rotation down on the Y axe (2→3), and finally a right rotation (3→4).

5 Conclusions and Future Work

In this work a tool for representing relations among concepts in 3D has been presented. The representation is based in molecular representations for chemical purposes. The developed algorithms first represent the concepts and relations in two dimensions and then they make a simulation of a 3D model.

We really think that this representation could be useful for search purposes, perhaps not for an inexpert user. It could be also useful as a representation of the conceptual relations in a certain domain, which are not usually specifically quantified.

Once the initial prototype of the tool has been developed, there would appear new necessities and applications with its daily and real use. It is sure that we will have to work in the integration of T-DiCoR in the framework that our team had been developing (tools such as FISS [1] or GUMSe [8]). For example, to define the XML formats of the interchange documents will be necessary. Another interesting improvement could be that the user can select a set of words (concepts), and the tool would generate the graph (or even when an user makes a query to our search engine, nowadays the user can only see the relations of words that are in the input matrix).

References

1. Olivas, J.A., Garcés, P.J., Romero, F.P. (2003). An application of the FIS-CRM model to the FISS metasearcher: Using fuzzy synonymy and fuzzy generality for representing concepts in documents, Int. Journal of Approximate Reasoning 34, 201-219.
2. Widyantoro, D., Yen, J. (2001). Incorporating fuzzy ontology of term relations in a search engine, Proceedings of the BISC Int. Workshop on Fuzzy Logic and the Internet, 155-160.
3. Olivas, J.A.; Puente, C.; Tejado, A. (2005). Searching for causal relations in text documents for ontological application. Proc. of the Int. Conf. on Artificial Intelligence IC-AI'05, Las Vegas, USA, CSREA Press, vol. II. 463 – 468.
4. Yager, R. R. (1988). On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. IEEE Transactions on Systems, Man and Cybernetics, vol. 18, nº 1, 183 - 190.
5. Castro, J.L.; Trillas, E.; Zurita, J. M. (1998). Non-monotonic Fuzzy Reasoning. Fuzzy Sets and Systems 94, North Holland, 217 - 225.
6. Harlow, E: 3d Java Molecule Viewer. In: http://www.netbrain.com/~brain/molecule/, 1999.
7. Fruchterman, T. M. J., Reingold, E. M. (1991). Graph Drawing by Force-directed Placement. In: Software – Practice & Experience, Vol. 21 (11): 1129-1164.
8. de la Mata, J.; Olivas, J. A.; Serrano-Guerrero, J. (2004). Improving Web Search by Using Query Customization with GUMSe, Current Issues on Data and Knowledge Engineering EUROFUSE'04, Warsaw, Poland, 325 – 333.