Towards Flexible Information Retrieval Based on CP-Nets

Fatiha Boubekeur[1,2], Mohand Boughanem[1], and Lynda Tamine-Lechani[1]

[1] IRIT-SIG, Paul Sabatier University,
31062 Toulouse, France
[2] Mouloud Mammeri University,
15000 Tizi-Ouzou, Algeria
boubekeu@irit.fr, boughane@irit.fr,
tamine@irit.fr

Abstract. This paper describes a flexible information retrieval approach based on CP-Nets (Conditional Preferences Networks). The CP-Net formalism is used for both representing qualitative queries (expressing user preferences) and representing documents in order to carry out the retrieval process. Our contribution focuses on the difficult task of term weighting in the case of qualitative queries. In this context, we propose an accurate algorithm based on UCP-Net features to automatically weight Boolean queries. Furthermore, we also propose a flexible approach for query evaluation based on a flexible aggregation operator adapted to the CP-Net semantics.

1 Introduction

The main goal of an information retrieval system (IRS) is to find the information assumed to be relevant to a user query generally expressed by a set of keywords (terms) connected with Boolean operators. However, keywords-based queries don't allow expressing user preferences on the search criteria. Furthermore, the classical Boolean aggregation is too "crisp" defining strict matching mechanisms. The traditional IRS, too rigid, thus provide only partial results and sometimes even non-relevant ones. To tackle these problems, various works focused on the extension of the classical Boolean model, by introducing weights in the query [1], [7], [8], [12], [10], [11], in order to enable a fuzzy representation of the documents and a flexible query formulation. More precisely, the Boolean model extensions proposed make possible the expression of the user preferences within the query using flexible aggregation operators; these operators are also applied on documents in order to allow flexible indexing and consequently flexible query evaluation. However, assigning weights to query terms is not an easy task for a user, particularly when the query contains conditional preferences. We illustrate the problem we attempt to solve using the following example.

Information need: "*I am looking for housing in Paris or Lyon of studios or university room type. Knowing that I prefer to be in Paris rather than to be in Lyon, if I should go to Paris, I will prefer being into residence hall (RH) (we will treat residence hall as a single term), whereas if I should go to Lyon, a studio is more*

H. Legind Larsen et al. (Eds.): FQAS 2006, LNAI 4027, pp. 222 – 231, 2006.

preferable to me than a room in residence hall. Moreover the Center town of Paris is more preferable to me than its suburbs; whereas if I must go to Lyon, I will rather prefer to reside in suburbs that in the center".

Such a query emphasizes conditional preferences. Taking into account the user preferences leads to the following query:

$$(Paris\ 0.9 \wedge (RH\ 0.6 \vee Studio\ 0.3) \wedge (Center\ 0.5 \vee Suburbs\ 0.4)) \vee$$
$$(Lyon\ 0.8 \wedge (RH\ 0.5 \vee Studio\ 0.8) \wedge (Center\ 0.7 \vee Suburbs\ 0.8)) .$$

In this representation, the weights of terms *R.H* and *Studio*, *Center* and *Suburbs*, are different when they are associated in *Paris* or *Lyon* respectively. This exactly expresses the conditional preferences of the user. The disjunctive normal form of this query is given by:

$$\begin{aligned}&(Paris\ 0.9 \wedge RH\ 0.6 \wedge Center\ 0.5) \vee (Paris\ 0.9 \wedge Studio\ 0.3 \wedge Center\ 0.5) \vee\\ &(Paris\ 0.9 \wedge RH\ 0.6 \wedge Suburbs\ 0.4) \vee (Paris\ 0.9 \wedge Studio\ 0.3 \wedge Suburbs\ 0.4)\\ &\vee (Lyon\ 0.8 \wedge RH\ 0.5 \wedge Center\ 0.7) \vee (Lyon\ 0.8 \wedge Studio\ 0.8 \wedge Center\ 0.7) \vee\\ &(Lyon\ 0.8 \wedge RH\ 0.5 \wedge Suburbs\ 0.8) \vee (Lyon\ 0.8 \wedge Studio\ 0.8 \wedge Suburbs\ 0.8) .\end{aligned} \tag{1}$$

Even thought this representation supports conditional preferences, nevertheless it poses problems. Indeed, assuming that each conjunctive sub-query of the whole query has a total importance weight, computed by aggregation of individual weights of its own terms (using *min* or *OWA* operators or simply by averaging for example), then we obtain: importance of $(Paris \wedge Studio \wedge Center)$ is *0.56* whereas importance of $(Lyon \wedge Studio \wedge Center)$ is *0.76* implying that the latest alternative is preferable than the preceding one. This is contradictory with the stated user preferences. Our weighting above is therefore incoherent.

This example outlines the impact of random or intuitive term weighting of a qualitative query on the semantic accuracy of the preferences it attempts to express. It illustrates the difficult task of query term weighting in a qualitative query. Other works in IR tackled this problem using more intuitive qualitative preferences, expressed with linguistic terms such: *important, very important*… [2], [3]. However, the problem of weighting terms belongs to the definition of both fuzzy concepts of importance and linguistic modifiers: *very, little…*

We propose, in this paper, a mixed approach for flexible IR which combines the expressivity property and the computation accuracy within a unified formalism: CP-Nets [4], [6]. More precisely, we propose to use the CP-Net formalism for two main reasons. The first one is to enable a graphical representation of flexible queries expressing user conditional preferences that can be automatically quantified using an accurate algorithm dedicated to UCP-Nets [5] valuation; such a quantification corresponds to the resolution of the problem of query term weighting presented above. The second reason is to allow a flexible query evaluation using a CP-Net document representation and a flexible matching mechanism based on the use of flexible aggregation operator adapted to the CP-Net semantics.

The paper is organized as follows: in section 2, we present the guiding principles of the CP-Nets and the UCP-Nets. The Section 3 describes our flexible information retrieval approach based on CP-Nets: we present namely our automatic CP-Nets weighting approach and our flexible CP-Net query evaluation method.

2 CP-Net Formalism

CP-Nets were introduced in 1999 [4] as graphical models for compact representation of qualitative preference relations. They exploit conditional preferential dependencies in the structuring of the user preferences under the C*eteris-Paribus*[1] assumption. Preference relations in a CP-Net can also be quantified with utility values leading to a UCP-Net. We describe in this section the CP-Nets and UCP-Nets graphical models.

2.1 CP-Nets

A CP-Net is a Directed Acyclic Graph, or *DAG*, $G = (V, E)$, where V is a set of nodes $\{X_1, X_2, X_3 ... X_n\}$ that represent the preference variables and E a set of directed arcs expressing preferential dependencies between them. Each variable X_i takes values in the set $Dom(X_i) = \{xi_1, x_{i2}, x_{i3}, \ldots\}$. We note $Pa(X_i)$ the parent set of X_i in G. representing his predecessor in the graph. A set $\{X_i, Pa(X_i)\}$ defines a CP-Net family.

For each variable X_i of the CP-Net, is attached a conditional preference table ($CPT(X_i)$) specifying for each value of $Pa(X_i)$ a total preference order among $Dom(X_i)$ values. For a root node of the CP-Net, the *CPT* simply specifies an unconditional preference order on its values.

Figure 1 illustrates the CP-Net corresponding to the query (1). The variables of interest are V={*City, Housing, Place}* where *Dom(City)={Paris, Lyon}*, *Dom(Housing) = {RH, Studio}* and *Dom (Place) = {Center, Suburbs}*. In addition, *CPT(City)* specifies that *Paris* is unconditionally preferable to *Lyon* (*Paris* $\succ$[2] *Lyon*), whereas *CPT(Housing)* for example, specifies a preference order on *Housing* values, under the condition of the *City* node values (thus for example, if *Paris* then *RH* $\succ$ *Studio*).

We call an alternative of the CP-Net each element of the Cartesian product of all its nodes values fields. It is interpreted like a conjunction of its elements. For example, *(Paris, Studio, Center)* and *(Lyon, RH, Center)* are alternatives of the CP-Net presented in figure 1.

A CP-Net induces a complete preference graph built on the whole of its alternatives, ordered under the *Ceteris Paribus* assumption [4]:

$$\text{Let } x_1, x_2 \in Dom(X),\ x_1 \succ x_2 \text{ Ceteris Paribus if } \forall p \in Pa(X),\ \forall y \in V\text{-}\{X, Pa(X)\}:$$
$$x_1\, p\, y \succ x_2\, p\, y\,.$$

The preference graph induced by the CP-Net of Figure 1 is presented in Figure 2, in which a directed arrow from X_i node (alternative) to X_j node expresses that X_j is preferable to X_i *Ceteris Paribus*. Hence, the alternatives *(Paris, RH, Center)* and *(Lyon, RH, Center)* are comparable (since *Paris* $\succ$ *Lyon Ceteris Paribus*), whereas, the two alternatives *(Paris, RH, Suburbs)* and *(Paris, Studio, Center)* are not and thus cannot be ordered.

---

[1] All else being equal.
[2] Preference relation.

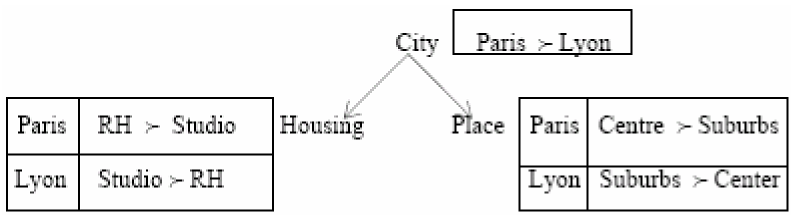

Fig. 1. CP-Net Representation of a Boolean query

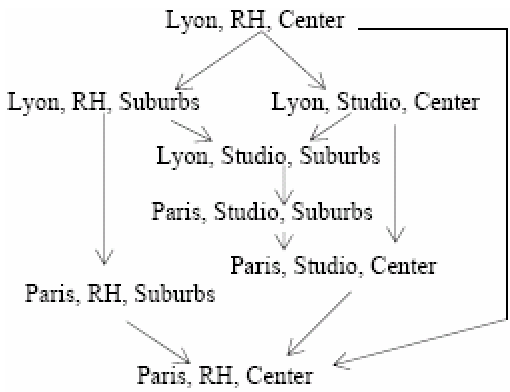

Fig. 2. A preference graph

2.2 UCP-Nets

A CP-Net doesn't allow comparison and ordering of all the possible alternatives. For this aim, one must quantify preferences. A UCP-Net [5] extends a CP-Net by quantifying the CP-Net nodes with conditional utility values (utility factors). A conditional utility factor $f_i(X_i, Pa(X_i))$ (we simply write $f_i(X_i)$), is a real value attached to each value X_i given an instantiation of its parents $Pa(X_i)$.

Therefore defining a UCP-Net amounts to define for each family $\{X_i, Pa(X_i)\}$ of the CP-Net, a utility factor $f_i(X_i)$. These factors are used to quantify the *CPTs* in the graph.

The utility factors are generalized additive independent (*GAI*) [5]. Formally, for a UCP-Net $G=(X, V)$ where $V=\{X_1,\ldots,X_n\}$, we compute the global utility of V denoted u(V) as follows:

$$u(V) = \sum_i f_i(X_i) . \quad (2)$$

In Figure 3, we present a UCP-Net that quantifies the CP-Net presented in Figure1.

For the UCP-Net of Figure 3, utility factors $f_1(City)$, $f_2(Housing, City)$ and $f_3(Place, City)$ being *GAI*, one has: $u(City, Housing, Place) = f_1(City) + f_2(Housing, City) + f_3(Place, City)$. This leads to the following: $u\ (Paris, Studio, Center) = 1.99$ and $u\ (Paris, RH, Suburbs) = 1.92$.

Consequently, one can argue that *(Paris, Studio, Center)* $\succ$ *(Paris, RH, Suburbs).* This clearly traduces an ordering of the preferences that couldn't be obtained using a basic CP-Net.

The validity of a UCP-Net is based on the principle of predominance [5] defined as follows: Let $G=(V, E)$ a quantified CP-Net. G is a valid UCP-Net if :

$$\forall X \in V, \quad Minspan\ (X) >= \sum_i Maxspan\ (Y_i)\ . \qquad (3)$$

Where Yi is a descendant of X ($Y_i \in V\ /\ X = Pa(Y_i)$) and

$$Minspan(X) = min_{\ x_1, x_2\ \in Dom(X)}\ (min_{\ p\in Dom(Pa(X))}\ (|\ f_X(x_1, p) - f_X(x_2, p)|))\ .$$

$$Maxspan(X) = max_{x1,\ x2\ \in Dom(X)}\ (max_{\ p\in Dom(Pa(X))}\ (|\ f_X\ (x_1, p) - f_X(x_2, p)|)).$$

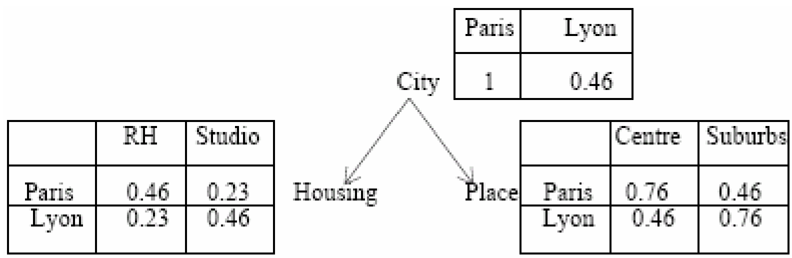

Fig. 3. An example of UCP-Net

3 Flexible Information Retrieval Based on CP-Nets

We describe in this section our flexible information retrieval approach based on CP-Nets. We show first of all how to use CP-Nets for expressing user qualitative queries, then we detail our approach for automatically weighting the related terms. We finally present our CP-Net semantics based flexible method for evaluating such preferential weighted queries.

3.1 Expressing Queries Using CP-Nets

The user preferences are expressed using concepts represented by variables. Each variable is defined on a domain of values (a value is therefore a query term). For each variable, the user must specify all of its preferential dependencies from which a CP-Net graph is built. The CP-Net query is then weighted by preference weights corresponding to utility factors. Our automatic weighting process is based on the predominance property stated above (section 2.2). We present it in the following:.

3.1.1 Weighting Queries Using UCP-Nets

Let $Q=(V,E)$ be a CP-Net query Q which expresses the qualitative conditional preferences of a user on n concepts (variables), X be a variable of Q, such as $|\ Dom(X)| = k$, and let $u(i)$ be the i^{th} preference order on X's values (one assume $u(i)$ growing when i grows):

For any leaf node X, we generate the utilities simply as uniform preference orders over the set [0 1] as follows:

$$u(1) = 0 \text{ and } u(i) = u(i-1) + (1/(k-1)), \quad \forall 1 < i <= k . \tag{4}$$

For any internal node X (X is not a leaf node), we compute $S = \sum_i Maxspan(B_i)$ where B_i represents the descendants of X. The predominance property (3) imposes that $Minspan(X) >= S$. Several values answer the condition correctly, the smallest one S is choosen, so that $Minspan(X) = S$. The utilities are computed as follows:

$$u(1) = 0 \text{ and } u(i) = u(i-1) + S, \quad \forall 1 < i <= k . \tag{5}$$

We then easily compute: $Minspan(X) = |u(i+1) - u(i)|$ *and* $Maxspan(X) = |u(k) - u(1)|$.

The utility values obtained can be higher than 1 (particularly in the case of internal nodes), we propose a normalisation of the individual utility factors of the CP-Net and of the total utilities of each alternative as follows: For each CP-Net node X_j, let $Max_{X_j} = max_i(u(i))$ be the highest preference order on X_j values, then:

$$\forall X_j, \ \forall u(i), \ 1 \le i \le |Dom(X_j)|, \ u(i) = u(i) / \textstyle\sum_j Max_{X_j} . \tag{6}$$

3.1.2 Illustration

Using the proposed method, the UCP-Net related to the query of Figure 1 is presented in Figure 4.

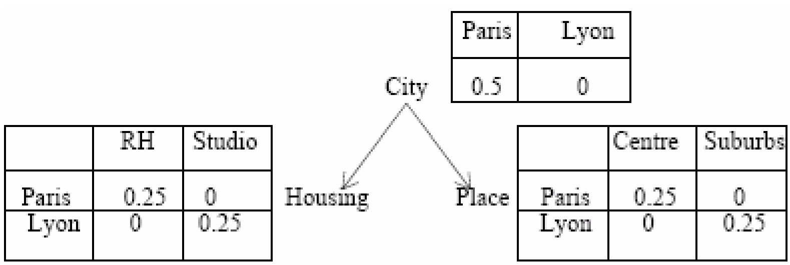

Fig. 4. A UCP-Net query

We thus obtain the following weighted Boolean query:

$$(Paris\ 0.5 \wedge (RH\ 0.25 \vee Studio\ 0) \wedge (Center\ 0.25 \vee Suburbs\ 0)) \vee (Lyon\ 0 \wedge (RH\ 0 \vee Studio\ 0.25) \wedge (Center\ 0 \vee Suburbs\ 0.25)$$

From where: $u(Paris, Studio, Center) = 0.5$ *and* $u(Paris, RH, Suburbs) = 0.75$.

3.2 CP-Net Based Query Evaluation

Once the CP-Net query weighted, the retrieval process is launched during the first step on the whole of the nodes values of the CP-Net without taking account of weighting as a preliminary. The result is a list of probable relevant documents for the query. In the second step, an evaluation process based on semantics CP-Net ranks them by degree of relevance; for this aim, retrieved documents are first represented by CP-Nets, then an evaluation process is proposed to estimate the relevance status values of such CP-Net documents for the CP-Net query.

3.2.1 A Document as a CP-Net

Each assumed relevant document for a query $Q=(V,E)$ is represented by a CP-Net $D=(V, E')$. The corresponding topology is similar to the query CP-Net $Q=(V,E)$ but the CPTs are different. Indeed, the related CPTs quantify the importance of indexing terms in D as estimated within the terms weights based on a variant of *tf* idf*.

The document (respectively the query) is then interpreted as a disjunction of conjunctions, each one of them being built on the whole of the elements of the Cartesian product $Dom(X_1)* Dom(X_2)*\ldots* Dom(X_n))$ where X_i $(1 \le i \le n)$ are the CP-Net document (respectively query) nodes, that is to say:

$$D = \vee_{j_i} (\wedge_i (t_{i,j_i}, p_{i,j_i})). \tag{7}$$

$$Q = \vee_{j_i} (\wedge_i (t_{i,j_i}, f_{i,j_i})) \tag{8}$$

Where $1 \le i \le n$, $1 \le j_i \le |Dom(X_i)|$, $t_{i,j_i} \in Dom(X_i)$, p_{i,j_i} is the weight of t_{i,j_i} in D (based on its occurrence frequency) and f_{i,j_i} is the weight of the term t_{i,j_i} (its utility) in Q being given a value of its parent.

Let us note $m = |Dom(X_1)| * |Dom(X_2)| *\ldots *|Dom(X_n)|$, by posing: $\wedge_i t_{i,j_i} = T_k$, with $1 \le k \le m$, the representations (7) and (8) are respectively brought to:

$$D = \vee_k (T_k, S_k) = \vee (T_k, S_k). \tag{9}$$

$$Q = \vee_k (T_k, U_k) = \vee (T_k, U_k). \tag{10}$$

Where S_k and U_k are the aggregate weights of values p_{i,j_i} respectively introduced in (7) and (8). S_k and U_k are computed as follows:

Computing U_k. Since the $f_{i,ji}$ factors are GAI, one has according to equality (2):

$$U_k = \Sigma_i f_{i,j_i}. \tag{11}$$

Computing S_k. We propose to compute the aggregated weight S_k, as weighted average of the p_{i,j_i} as follows: We first associate an importance of position G_X to nodes X of the CP-Net document according to their levels in the graph: if X is a leaf node: $G_X=1$; for any other node X such as B_l are the descendants of X and G_{B_l} their respective importance orders, one has:

$$G_X = max_l\ G_{B_l} + 1. \tag{12}$$

The aggregate weight S_K introduced in (9) is then given by:

$$S_K = \Sigma_i p_{i,j_i} G_{X_i} / \Sigma_i G_{X_i}. \tag{13}$$

Where X_i is the node containing term (t_{i,j_i}) of D.

D_1 *((Paris, 0.7), (Lyon, 0.5), (RH, 0.2))* .

Fig. 5. Retrieved document

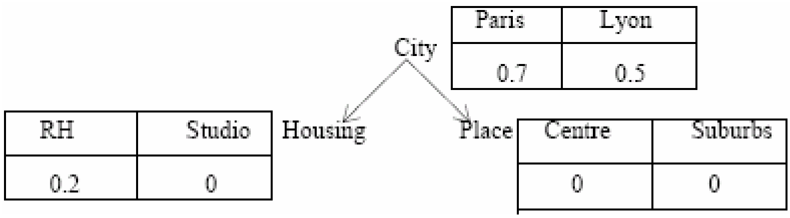

Fig. 6. D_1 as CP-Net

Thus for example, let us suppose that the retrieval process, launched initially on the whole terms of the weighted query of Figure 4, returns document D_1 presented in Figure 5, where each pair *(t, p)* respectively represents the term and its weight associated in the document. In Figure 6 we present the UCP-Net associated with documents D_1.

UCP-Net query introduced in Figure 4 and UCP-Net document introduced in Figure 6 are interpreted respectively using formulas (7) and (8), as follows:

$$Q = ((Paris,0.5) \wedge (RH,0.25) \wedge (Center,0.25)) \vee ((Paris,0.5) \wedge (RH,0.25) \wedge (Suburbs,0)) \vee ((Paris,0.5) \wedge (Studio,0) \wedge (Center,0.25)) \vee ((Paris,0.5) \wedge (Studio,0) \wedge (Suburbs,0)) \vee ((Lyon,0) \wedge (RH,0) \wedge (Center,0)) \vee ((Lyon,0) \wedge (RH,0) \wedge (Suburbs,0.25)) \vee ((Lyon,0) \wedge (Studio,0.25) \wedge (Center,0)) \vee ((Lyon,0) \wedge (Studio,0.25) \wedge (Suburb,0.25)) .$$

$$D_1 = ((Paris, 0.7) \wedge (RH, 0.2) \wedge (center, 0)) \vee ((Paris, 0.7) \wedge (RH, 0.2) \wedge (suburbs,0)) \vee ((Paris, 0.7) \wedge (Studio, 0) \wedge (center, 0)) \vee ((Paris, 0.7) \wedge (Studio, 0) \wedge (suburbs,0)) \vee ((Lyon, 0.5) \wedge (RH, 0.2) \wedge (center, 0)) \vee ((Lyon, 0.5) \wedge (RH, 0.2) \wedge (suburbs,0)) \vee (Lyon, 0.5) \wedge (Studio, 0) \wedge (center, 0)) \vee ((Lyon, 0.5) \wedge (Studio, 0) \wedge (suburbs,0)) .$$

Thus, the CP-Net query Q presented in figure 4 and the CP-Net document in Figure 6 are translated respectively according to formulas (9) and (10) into:

$$Q = (T_1, 1) \vee (T_2 ,0.75) \vee (T_3 ,0.75) \vee (T_4 , 0.5) \vee (T_5 ,0) \vee (T_6 ,0.25) \vee (T_7 , 0.25) \vee (T_8 , 0.5) .$$

$$D_1= (T_1, 0.4) \vee (T_2,0.4) \vee (T_3, 0.35) \vee (T_4, 0.35) \vee (T_5,0.3) \vee (T_6, 0.3) \vee (T_7,0.25) \vee (T_8, 0.25) .$$

Where T_i, $1 \leq i \leq 8$ is given in Table 1, T_i's weight in Q (respectively in D_1) is computed using (11) (respectively (13)).

Table 1. Conjunctive sub-queries

T_1= (Paris $\wedge$ RH $\wedge$ Center)	T_2 = (Paris $\wedge$ RH $\wedge$ Suburbs)
T_3= (Paris $\wedge$ Studio $\wedge$ Center)	T_4 = (Paris $\wedge$ Studio $\wedge$ Suburbs)
T_5 = (Lyon $\wedge$ RH $\wedge$ Center)	T_6 = (Lyon $\wedge$ RH $\wedge$ Suburbs)
T_7= (Lyon $\wedge$ Studio $\wedge$ Center)	T_8= (Lyon $\wedge$ Studio $\wedge$ Suburbs)

3.2.2 Query Evaluation

Let Q be a CP-Net query expressed as in (10), and D the retrieved document expressed as in (9). In order to evaluate the relevance of the document D for the weighted query Q, $RSV(Q,D)$, we propose to adapt and use the weighted minimum operator [12], [9] as follows:

Let U_K be the importance weight of T_k in Q, $F(D, T_k) = S_K$, the weight of T_k in the document D, we note $RSV_{T_k}(F(D, T_k), U_k)$ the evaluation function of T_k for document D. The various weighted conjunctions (T_k, U_k) being bound by a disjunction, which gives:

$$RSV_{T_k}(F(D, T_k), U_k) = Min\ (S_k\ U_k). \tag{14}$$

$RSV(Q,d)$ is then obtained by aggregation of the whole of the weights of relevance computed in (14) as follows:

$$RSV(Q,D) = Max_K(Min\ (S_k, U_k)). \tag{15}$$

Using the Equalities (14) and (15), we compute the partial relevance of document D_1 for each sub-query T_k given in Table 1 and its total relevance for the disjunctive query Q as indicated in Table 2 below:

Table 2. Partial and total relevance of document D_1

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	GRSV[3]
D_1	0.4	0.4	0.35	0.35	0	0.25	0.25	0.25	0.4

Document D_1 can thus be ordered either partially according to its partial relevance for each sub-query T_k, or globally according to its total relevance to query $Q = \vee T_k$.

4 Conclusion

We described in this paper a novel approach for flexible IR based on CP-Nets. The approach focuses on the representation of qualitative queries expressing user preferences. The formalism is graphic and qualitative what allows a natural and intuitive formulation and a simple and compact representation of the preferences. The qualitative formalism has a power of high expression but declines in computing

[3] Global RSV.

power. We proposed then to quantify it using utility values leading to a UCP-Net. The utilities, representing conditional importance weights of query terms, are computed automatically. The user is thus discharged from this tiresome and not less improbable task, and the generated weights are checked correct since based on theoretical bases of UCP-Nets. We also proposed a CP-Net based query evaluation. Our approach aims to represent retrieved document as CP-Net in order to estimate both its partial relevance and its total relevance to a given query by using of a flexible aggregation operator, the weighted minimum, which we adapted to CP-Nets semantics.

One interesting future work is the use of fuzzy concepts and linguistic terms in the user's need expression and their integration into the CP-Net semantics.

It would be either interesting to improve our approach by taking into account partial preference relations that could be expressed by the user.

References

1. Bordogna, G., Carrara, P., Pasi, G.: Query term weights as constraints in fuzzy information retrieval. Information Processing and Management, 27(1) (1991) 15-26.
2. Bordogna, G., Pasi, G.: A fuzzy linguistic approach generalizing Boolean information retrieval: a model and its evaluation. Journal of the American Society for Information Science, 44(2) (1993) 70-82.
3. Bordogna, G., Pasi, G.: Linguistic aggregation operators of selection criteria in fuzzy information retrieval. International Journal of Intelligent Systems, (10) (1995) 233-248.
4. Boutilier, C., Brafman, R., Hoos, H., Poole, D.: Reasoning with Conditional Ceteris Paribus Preference Statements. In Proc. of UAI, (1999) 71 –80.
5. Boutilier, C., Bacchus, F., Brafman, R.: UCP-networks: A directed graphical representation of conditional utilities. In Proc. of UAI, (2001) 56 –64.
6. Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., Poole, D.: CP-Nets A tool for representing and reasoning about conditional ceteris paribus preference statements. Journal of Artificial Research Intelligence, (21) (2004) 135 –191.
7. Buell, D. A., Kraft, D. H.: A model for a weighted retrieval system. Journal of the American Society for Information Science, 32(3) (1981) 211-216,.
8. Crestani, F., Pasi, G.: Soft information retrieval: Applications of fuzzy Set Theory and Neural Networks. In "Neuro Fuzzy Techniques For Intelligent Information Systems ". N Kasabov and Robert Kozmz Editors, Physica –Verlag, Springer-Verlag Group, (1999) 287-313.
9. Dubois, D., Prade, H.: Weighted minimum and maximum operations in fuzzy set theory. Information Sciences, (39) (1986) 205-210.
10. Kraft, D. H., Buell, D.: Fuzzy sets and generalized Boolean retrieval systems. International Journal of Man-Machine Studies, 19(1) (1983) 45-56.
11. Pasi, G. :A logical formulation of the Boolean model and of weighted Boolean model. In Proceedings of the Workshop on Logical and Uncertainty Models for Information Systems, London, U.K (1999) 1 –11.
12. Yager, R.: A note on weighted queries in information retrieval systems. In Journal of American Society for Information Science, 38(1) (1987) 23-24.