The Garnata Information Retrieval System at INEX'07

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete,
Carlos Martín-Dancausa, and Alfonso E. Romero

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática y de Telecomunicación, Universidad de Granada,
18071 – Granada, Spain
{lci,jmfluna,jhg,cmdanca,aeromero}@decsai.ugr.es

Abstract. This paper exposes the results of our participation at INEX'07 in the AdHoc track and the comparison of these results with respect to the ones obtained last year. Three runs were submitted to each of the Focused, Relevant In Context and Best In Context tasks, all of them obtained with Garnata, our Information Retrieval System for structured documents. As in the past year, we use a model based on Influence Diagrams, the CID model. The result of our participation has been better than the last year and we have reached an acceptable position in the ranking for the three tasks. In the paper we describe the model, the system and we show the differences between our systems at INEX'06 and INEX'07, which make possible to get a better performance.

1 Introduction

This is the second year that members of the research group "Uncertainty Treatment in Artificial Intelligence" at the University of Granada submit runs to the INEX official tasks, although before 2006 we also contributed to INEX with the design of topics and the assessment of relevance judgements. Like in the past year, we have participated in the Ad hoc Track with an experimental platform to perform structured retrieval using Probabilistic Graphical Models [5,8,10], called Garnata [4].

This year we have improved the version of Garnata that we used at INEX'06 in two ways, and we have also adapted it to cope with the three, non thorough tasks proposed this year, namely Focused, Relevant in Context and Best in Context. For each of these tasks, we have submitted three runs, all of them using Garnata with a different set of parameters. The results of this second participation are considerably better than those of the past year, where we were in the last positions of the ranking. Nevertheless, we are still quite far from the first positions, so there is still room for improvement, and more research and experimentation need to be carried out.

The paper is organised as follows: the next section describes the probabilistic graphical models underlying Garnata. Sections 3 and 4 give details about the new characteristics/improvements incorporated into the system and the adaptation

N. Fuhr et al. (Eds.): INEX 2007, LNCS 4862, pp. 57–69, 2008.

of Garnata to generate outputs valid for the three tasks, respectively. In Section 5 we discuss the experimental results. The paper ends with the conclusions and some proposals for future work with our system.

2 Probabilistic Graphical Models in the Garnata System

The Garnata IRS is based on probabilistic graphical models, more precisely an influence diagram and the corresponding underlying Bayesian network. In this section we shall describe these two models and how they are used to retrieve document components from a document collection through probabilistic inference (see [2,3] for more details). Alternative probabilistic graphical models for structured information retrieval can also be found in the literature [6,7,9]. We assume a basic knowledge about graphical models.

2.1 The Underlying Bayesian Network

We consider three different kinds of entities associated to a collection of structured documents, which are represented by the means of three different kinds of random variables: *index terms*, *basic structural units*, and *complex structural units*. These variables are in turn represented in the Bayesian network through the corresponding *nodes*. Term nodes form the set $\mathcal{T} = \{T_1, T_2, \ldots, T_l\}$; $\mathcal{U}_b = \{B_1, B_2, \ldots, B_m\}$ is the set of basic structural units, those document components which only contain terms, whereas $\mathcal{U}_c = \{S_1, S_2, \ldots, S_n\}$ is the set of complex structural units, that are composed of other basic or complex units. For those units containing both text and other units, we consider them as complex units, and the associated text is assigned to a new basic unit called *virtual unit*, see the example in Figure 1[1]. The set of all structural units is therefore $\mathcal{U} = \mathcal{U}_b \cup \mathcal{U}_c$.

The binary random variables associated with each node T, B or S take its values from the sets $\{t^-, t^+\}$, $\{b^-, b^+\}$ or $\{s^-, s^+\}$ (the term/unit is not relevant or is relevant), respectively. A unit is considered relevant for a given query if it satisfies the user's information need expressed by this query. A term is relevant in the sense that the user believes that it will appear in relevant units/documents.

Regarding the arcs of the model, there will be an arc from a given node (either term or structural unit) to the particular structural unit the node belongs to. The hierarchical structure of the model determines that each structural unit $U \in \mathcal{U}$ has *only one* structural unit as its child: the unique structural unit containing U (except for the leaf nodes, i.e. the complete documents, which have no child). We shall denote $U_{hi(U)}$ the single child node associated with node U (with $U_{hi(U)} =$ null if U is a leaf node).

To assess the numerical values for the required probabilities $p(t^+)$, $p(b^+ |pa(B))$ and $p(s^+|pa(S))$, for every node in $\mathcal{T}$, $\mathcal{U}_b$ and $\mathcal{U}_c$, respectively, and every

[1] Of course this type of unit is non-retrievable and it will not appear in the XPath route of its descendants, it is only a formalism that allows us to clearly distinguish between units containing only text and units containing only other units.

```
<section>
<title>t1 t2 t3 t4 t7</title>
t3 t4 t5 t6
<normallist>
<item>t4 t7 t9 t10 t11</item>
<item>t6 t8 t11</item>
</normallist>
</section>
```

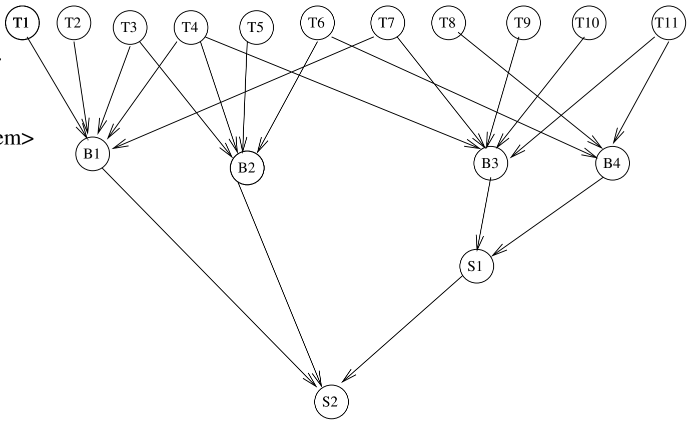

Fig. 1. Sample XML text and the corresponding Bayesian network. Ti represent index terms; the basic unit B1 corresponds with the tag `<title>`, and B3 and B4 with the tag `<item>`; the complex units S1 and S2 correspond with the tags `<normallist>` and `<section>` respectively; B2 is a virtual unit used to store the text within S2 which is not contained in any other unit inside it.

configuration $pa(X)$ of the corresponding parent sets $Pa(X)$, we use the canonical model proposed in [1], which supports a very efficient inference procedure. These probabilities are defined as follows:

$$\forall B \in \mathcal{U}_b, \quad p(b^+|pa(B)) = \sum_{T \in R(pa(B))} w(T, B)\,, \tag{1}$$

$$\forall S \in \mathcal{U}_c, \quad p(s^+|pa(S)) = \sum_{U \in R(pa(S))} w(U, S)\,, \tag{2}$$

where $w(T, B)$ is a weight associated to each term T belonging to the basic unit B and $w(U, S)$ is a weight measuring the importance of the unit U within S. In any case $R(pa(U))$ is the subset of parents of U (terms for B, and either basic or complex units for S) relevant in the configuration $pa(U)$, i.e., $R(pa(B)) = \{T \in Pa(B) \,|\, t^+ \in pa(B)\}$ and $R(pa(S)) = \{U \in Pa(S) \,|\, u^+ \in pa(S)\}$. These weights can be defined in any way with the only restrictions that

$$w(T, B) \geq 0, \quad w(U, S) \geq 0, \sum_{T \in Pa(B)} w(T, B) \leq 1, \text{ and } \sum_{U \in Pa(S)} w(U, S) \leq 1.$$

2.2 The Influence Diagram Model

The Bayesian network is now enlarged by including decision nodes, representing the possible alternatives available to the decision maker, and utility nodes, thus transforming it into an influence diagram. For each structural unit $U_i \in \mathcal{U}$,

R_i represents the decision variable related to whether or not to return U_i to the user (with values r_i^+ and r_i^-, meaning 'retrieve U_i' and 'do not retrieve U_i', respectively), and the utility node V_i measures the value of utility for the corresponding decision. We shall also consider a *global utility node* Σ representing the joint utility of the whole model (we assume an additive behavior of the model).

In addition to the arcs between the nodes present in the Bayesian network, a set of arcs pointing to utility nodes are also included, employed to indicate which variables have a direct influence on the desirability of a given decision. In order to represent that the utility function of V_i obviously depends on the decision made and the relevance value of the structural unit considered, we use arcs from each structural unit node U_i and decision node R_i to the utility node V_i. Moreover, we include also arcs going from $U_{hi(U_i)}$ to V_i, which represent that the utility of the decision about retrieving the unit U_i also depends on the relevance of the unit which contains it (of course, for those units U where $U_{hi(U)} =$ null, this arc does not exist). The utility functions associated to each utility node V_i are therefore $v(r_i, u_i, u_{hi(U_i)})$, with $r_i \in \{r_i^-, r_i^+\}$, $u_i \in \{u_i^-, u_i^+\}$, and $u_{hi(U_i)} \in \{u_{hi(U_i)}^-, u_{hi(U_i)}^+\}$.

Finally, the utility node Σ has all the utility nodes V_i as its parents. These arcs represent the fact that the joint utility of the model will depend on the values of the individual utilities of each structural unit. Figure 2 displays the influence diagram corresponding to the previous example.

2.3 Inference and Decision Making

Our objective is, given a query, to compute the expected utility of retrieving each structural unit, and then to give a ranking of those units in decreasing order of expected utility (at this moment we assume a thorough task, i.e. structural units in the output may overlap. In Section 4 we shall see how overlapping may be removed). Let $\mathcal{Q} \subseteq \mathcal{T}$ be the set of terms used to express the query. Each term $T_i \in \mathcal{Q}$ will be instantiated to t_i^+; let q be the corresponding configuration of the variables in $\mathcal{Q}$. We wish to compute the expected utility of each decision given q. As we have assumed a global additive utility model, and the different decision variables R_i are not directly linked to each other, we can process each one independently. The expected utilities for retrieving each U_i can be computed by means of:

$$EU(r_i^+ \mid q) = \sum_{\substack{u_i \in \{u_i^-, u_i^+\} \\ u_{hi(U_i)} \in \left\{u_{hi(U_i)}^-, u_{hi(U_i)}^+\right\}}} v(r_i^+, u_i, u_{hi(U_i)})\, p(u_i, u_{hi(U_i)}|q) \qquad (3)$$

Although the bidimensional posterior probabilities $p(u_i, u_{hi(U_i)}|q)$ in eq. (3) could be computed exactly, it is much harder to compute them that the unidimensional posterior probabilities $p(u_i|q)$, which can be calculated very efficiently due to the specific characteristics of the canonical model used to define the conditional probabilities and the network topology. So, we approximate the bidimensional

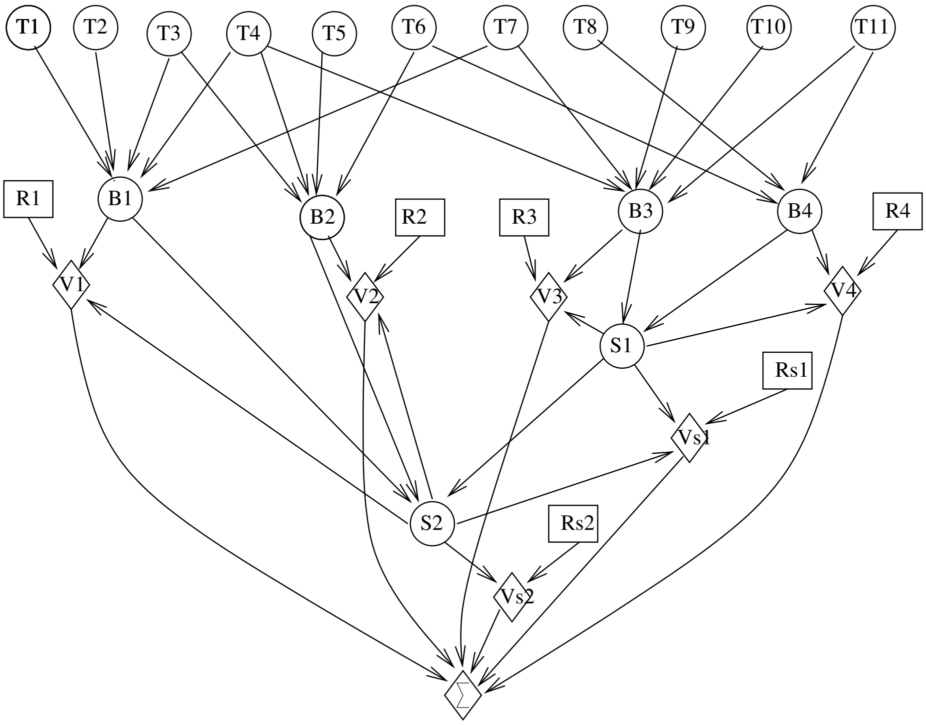

Fig. 2. Influence diagram for the example in Figure 1

probabilities as $p(u_i, u_{hi(U_i)}|q) = p(u_i|q) \times p(u_{hi(U_i)}|q)$. The computation of the unidimensional probabilities is based on the following formulas [2,3]:

$$\forall B \in \mathcal{U}_b, \qquad p(b^+|q) = \sum_{T \in Pa(B) \setminus Q} w(T,B)\, p(t^+) + \sum_{T \in Pa(B) \cap R(q)} w(T,B)\,, \quad (4)$$

$$\forall S \in \mathcal{U}_c, \qquad p(s^+|q) = \sum_{U \in Pa(S)} w(U,S)\, p(u^+|q)\,. \quad (5)$$

Figure 3 shows an algorithm that efficiently computes these probabilities, derived from eqs. (4) and (5), traversing only the nodes of the graph that require updating. It is assumed that the prior probabilities of all the nodes are stored in prior[X]; the algorithm uses variables prob[U] which, at the end of the process, will store the corresponding posterior probabilities. Essentially, the algorithm starts from the terms in $\mathcal{Q}$ and carries out a width graph traversal until it reaches the basic units that require updating, thus computing $p(b^+|q)$. Then, starting from these modified basic units, it carries out a depth graph traversal to compute $p(s^+|q)$, only for those complex units that require updating.

```
for each item T in Q
      for each unit B child of T
          if (prob[B] exists)
              prob[B] += w(T,B)*(1-prior[T]);
          else { create prob[B];
                    prob[B] = prior[B]+w(T,B)*(1-prior[T]); }
for each basic unit B s.t. prob[B] exists {
   U = B; prod = prob[B]–prior[B];
   while ($U_{hi(U)}$ is not NULL) {
      S = $U_{hi(U)}$;
      prod *= w(U,S);
      if (prob[S] exists)
         prob[S] += prod;
      else { create prob[S];
               prob[S] = prior[S]+prod; }
      U = S; }
 }
```

Fig. 3. Computing $p(b^+|q)$ and $p(s^+|q)$

The algorithm that initialises the process by computing the prior probabilities prior[U] (as the terms $T \in \mathcal{T}$ are root nodes, the prior probabilities prior[T] do not need to be calculated, they are stored directly in the structure) is quite similar to the previous one, but it needs to traverse the graph starting from all the terms in $\mathcal{T}$.

3 Changes from the Model Presented at INEX 2006

The two changes with respect to the model used at INEX'06 are related to the parametric part of the Garnata model. We explain first these parameters uses at INEX'06, before describing the changes made.

3.1 Parameters in Garnata

The parameters that need to be fixed in order to use Garnata are the prior probabilities of relevance of the terms, $p(t^+)$, the weights $w(T, B)$ and $w(U, S)$ used in eqs. (4) and (5), and the utilities $v(r_i^+, u_i, u_{hi(U_i)})$.

For the prior probabilities Garnata currently uses an identical probability for all the terms, $p(t^+) = p_0$, $\forall T \in \mathcal{T}$, with $p_0 = \frac{1}{|\mathcal{T}|}$.

The weights of the terms in the basic units, $w(T, B)$, follow a normalized tf-idf scheme:

$$w(T, B) = \frac{tf(T, B) \times idf(T)}{\sum_{T' \in Pa(B)} tf(T', B) \times idf(T')} \tag{6}$$

The weights of the units included in a complex unit, $w(U,S)$, measure, to a certain extent, the proportion of the content of the unit S which can be attributed to each one of its components:

$$w(U,S) = \frac{\sum_{T \in An(U)} tf(T, An(U)) \times idf(T)}{\sum_{T \in An(S)} tf(T, An(S)) \times idf(T)} \tag{7}$$

where $An(U) = \{T \in \mathcal{T} \mid T \text{ is an ancestor of } U\}$, i.e., $An(U)$ is the set of terms that are included in the structural unit U.

The utilities which are necessary to compute the expected utility of retrieving structural units, $EU(r_i^+ \mid q)$, namely $v(r_i^+, u_i, u_{hi(U_i)})$, are composed of a component which depends on the involved unit and another component independent on the specific unit and depending only on which one of the four configurations, $(u_i^-, u_{hi(U_i)}^-)$, $(u_i^-, u_{hi(U_i)}^+)$, $(u_i^+, u_{hi(U_i)}^-)$ or $(u_i^+, u_{hi(U_i)}^+)$, is being considered:

$$v(r_i^+, u_i, u_{hi(U_i)}) = nidf_Q(U_i) \times v(u_i, u_{hi(U_i)}) \tag{8}$$

with $v(u_i^-, u_{hi(U_i)}^-) = v^{--}$, $v(u_i^-, u_{hi(U_i)}^+) = v^{-+}$, $v(u_i^+, u_{hi(U_i)}^-) = v^{+-}$ and $v(u_i^+, u_{hi(U_i)}^+) = v^{++}$.

The part depending on the involved unit is defined as the sum of the inverted document frequencies of those terms contained in U_i that also belong to the query Q, normalized by the sum of the idfs of the terms contained in the query (a unit U_i will be more useful, with respect to a query Q, as more terms indexing U_i also belong to Q):

$$nidf_Q(U_i) = \frac{\sum_{T \in An(U_i) \cap Q} idf(T)}{\sum_{T \in Q} idf(T)} \tag{9}$$

Regarding the other component of the utility function independent on the involved unit, at INEX 2006 we used the following values

$$v^{--} = v^{-+} = v^{++} = 0\,,\ v^{+-} = 1$$

3.2 Changing Weights

We have modified the weights of the units included in a complex unit, $w(U,S)$, in order to also take into account, not only the proportion of the content of S which is due to U, but also some measure of the importance of the type (tag) of unit U within S. For example, the terms contained in a `collectionlink` (generally proper nouns and relevant concepts) or `emph2` should be cuantified higher than terms outside those units. Units labeled with `title` are also very informative, but units with `template` are not.

So, we call $I(U)$ the *importance of the unit* U, which depends of the type of tag associated to U. These values constitute a global set of free parameters, specified at indexing time. The new weights $nw(U,S)$, are then computed from the old ones in the following way:

$$nw(U,S) = \frac{I(U) \times w(U,S)}{\sum_{U' \in Pa(S)} I(U') \times w(U',S)} \tag{10}$$

Table 1. Importance of the different types of units used in the official runs

Tag	Weight file 8	Weight file 11	Weight file 15
name	20	100	200
title	20	50	50
caption	10	10	30
collectionlink	10	10	30
emph2	10	30	30
emph3	10	30	30
conversionwarning	0	0	0
languagelink	0	0	0
template	0	0	0

Table 2. Relative utility values of the different types of units used in the official runs

Tag	Utility file 1	Utility file 2	Utility file 3
conversionwarning	0	0	0
name	0.75	0.75	0.85
title	0.75	0.75	0.85
collectionlink	0.75	1.5	0.75
languagelink	0	0	0
article	2	2.5	2.5
section	1.5	1	1.25
p	1.5	1	1.5
body	1.5	1	2
emph2	1	1.5	1
emph3	1	1.5	1

We show in Table 1 the three different importance schemes used in the official runs. Unspecified importance values are set to 1 (notice that by setting $I(U) = 1, \forall U \in \mathcal{U}$, we get the old weights).

3.3 Changing Utilities

This year the formula of the utility values for a unit U is computed by considering another factor called *relative utility value*, $RU(U)$, which depends only on the kind of tag associated to that unit, so that:

$$v(r_i^+, u_i, u_{hi(U_i)}) = nidf_Q(U_i) \times v(u_i, u_{hi(U_i)}) \times RU(U_i) \tag{11}$$

It should be noticed that this value $RU(U)$ is different from the importance $I(U)$: a type of unit may be considered very important to contribute to the relevance degree of the unit containing it and, at the same time, is considered not very useful to retrieve this type of unit itself. For example, this may be the case of units having the tag `<title>`: in general a title alone may be not very useful for a user as the answer to a query, probably the user would prefer to get the content of the structural unit having this title; however, terms in a title tends to be highly representative of the content of a document part, so that the importance of the title should be greater than the importance derived simply of the proportion of text that the title contains (which will be quite low). The sets of utility values used in the official runs are displayed in Table 2.

In all the cases, the default value for the non-listed units is 1.0. We have also considered the case where all the relative utility values are set to 1.0 (which is equivalent to not to use relative utilities at all).

4 Adapting Garnata to the INEX 2007 Ad Hoc Retrieval Tasks

For each query, Garnata generates a list of document parts or structural units, ordered by relevance value (expected utility), as the output. So, this output is compatible with the thorough task used in previous editions but not with the three adhoc tasks for INEX 2007, *Focused*, *Relevant in Context* and *Best in Context*. To cope with these tasks, we still use Garnata but after we filter its output in a way which depends on the kind of task:

Focused task: The output must be an ordered list of structural units where overlapping has been eliminated. So, we must supply some criterion to decide, when we find two overlapping units in the output generated by Garnata, which one to preserve in the final output. The criterion we have used is to keep the unit having the greatest relevance value and, in case of tie, we keep the more general unit (the one containing a larger amount of text).

Relevant in Context task: In this case the output must be an ordered list of documents and, for each document, a set of non-overlapping structural units, representing the relevant text within the document (i.e., a list of non-overlapping units clustered by document). Therefore, we have to filter the output of Garnata using two criteria: how to select the non-overlapping units for each document, and how to rank the documents. To manage overlapping units we use the same criterion considered for the focused task. To rank the documents, we have considered three criteria to assign a relevance value to the entire document: the relevance value of a document is equal to: (1) the maximum relevance value of its units; (2) the relevance value of the "/article[1]" unit; (3) the sum of the relevance values of all its units. Some preliminary experimentation pointed out that the maximum criterion performed better, so we have used it in the official runs.

Best in Context task: The output must be an ordered list composed of a single unit per document. This single document part should correspond to the best entry point for starting to read the relevant text in the document. Therefore, we have to provide a criterion to select one structural unit for each document and another to rank the documents/selected units. This last criterion is the same considered in the relevant in context task (the maximum relevance value of its units). Regarding the way of selecting one unit per document, the idea is to choose some kind of *centroid* structural unit: for each unit U_i we compute the sum of the distances from U_i to each of the other units U_j in the document, the distance between U_i and U_j being measured as the number of links in the path between units U_i and U_j in the XML tree times the relevance value of unit U_j; then we select the unit having minimum sum of distances. In this way we try to select a unit which is nearest to the units having high relevance values.

5 Results of Our Model at INEX 2007

We have obtained the following results in the three tasks, using the combinations of weight and utility configurations displayed in Tables 3, 4 and 5.

As we can see in these results, the configuration of utilities with the value 3 is the most appropriate to get the best results in the different tasks, although we can not fix a specific configuration of weights that obtain the same results.

Finally, we show the graphics of the different tasks, where we can see the comparison of our results (red lines) with the results of the other organizations.

We have come to the conclusion that our system gets better results than the year before, so we have reached a middle position in the ranking (except for the focused task, where the results are worse) as we can see in the graphics and in the tables.

Table 3. Results for the Focused task

Weight file	Utility file	Ranking
8	3	62/79
15	2	70/79
15	none	71/79

Table 4. Results for the Relevant in Context task

Weight file	Utility file	Ranking
15	3	44/66
8	3	45/66
11	1	47/66

Table 5. Results for the Best in Context task

Weight file	Utility file	Ranking
8	3	40/71
15	None	45/71
15	2	47/71

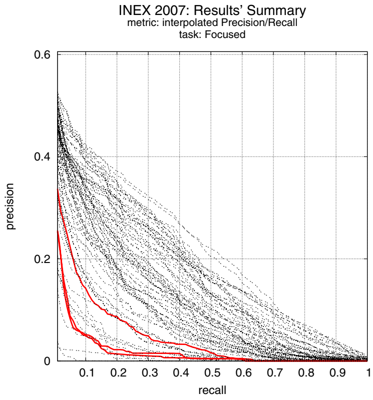

Fig. 4. Results for the Focused task

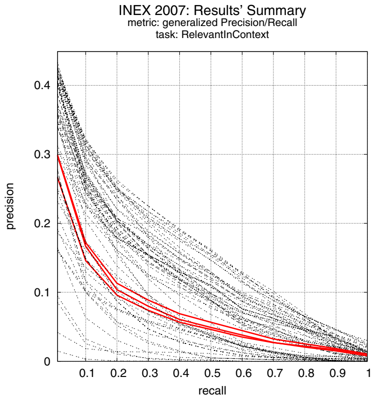

Fig. 5. Results for the Relevant in Context task

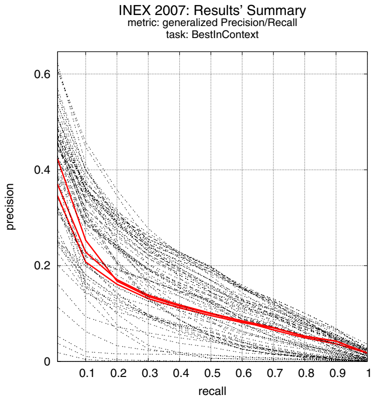

Fig. 6. Results for the Best in Context task

6 Concluding Remarks

In this year, our participation in the AdHoc track has been more productive than the one presented last year. In 2006, we only applied for one of the four AdHoc tasks (Thorough), and in 2007 we have sent results for all the tasks of the track. Besides, on 2006 we got a very bad ranking (lying on the percentile 91). The best runs of this year are clearly better than the one obtained last year (corresponding to percentiles 78 [Focused], 66 [Relevant in Context] and 56 [Best in Context]).

Results in the Relevant in Context and Best in Context tasks are at the end of the second-third of the ranking, but in Focused they are in a lower position. So, the filter used for Focused should be considerably improved.

On the other hand, we have not done yet a deep experimentation of different configurations for both the importance and the utility values. The parameters values used during INEX'07 were randomly selected configurations that obtained good results with the queries and relevance assessments of INEX'06. We think that the behaviour of our model could be clearly improved with a more systematic experimentation finding an optimal configuration of the parameters.

Acknowledgments. This work has been jointly supported by the Spanish Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía, Ministerio de Educación y Ciencia and the research programme Consolider Ingenio 2010, under projects TIC-276, TIN2005-02516 and CSD2007-00018, respectively.

References

1. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F.: The BNR model: foundations and performance of a Bayesian network-based retrieval model. Int. J. Appr. Reason. 34, 265–285 (2003)
2. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F.: Using context information in structured document retrieval: An approach using Influence diagrams. Inform. Process. Manag. 40(5), 829–847 (2004)
3. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F.: Improving the context-based influence diagram for structured retrieval. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 215–229. Springer, Heidelberg (2005)
4. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Romero, A.E.: Garnata: An information retrieval system for structured documents based on probabilistic graphical models. In: Proceedings of the Eleventh International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), pp. 1024–1031 (2006)
5. Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer, Heidelberg (2001)
6. Metzler, D., Croft, W.B.: Combining the language model and inference networks approaches to retrieval. Inform. Process. Manag. 40(5), 735–750 (2004)
7. Myaeng, S.H., Jang, D., Kim, M., Zhoo, Z.: A flexible model for retrieval of SGML documents. In: Proceedings of the 21st ACM–SIGIR Conference, pp. 138–145. ACM Press, New York (1998)

8. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan and Kaufmann, San Mateo (1988)
9. Piwowarski, B., Gallinari, P.: A Bayesian network for XML information retrieval: searching and learning with the INEX collection. Inform. Retrieval 8(4), 655–681 (2005)
10. Shachter, R.: Probabilistic inference and influence diagrams. Oper. Res. 36(5), 527–550 (1988)